

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**

L Number	Hits	Search Text	DB	Time stamp
1	7	(((((FSM or IFSM or (finite adj state adj machine)) and simulat\$4) and network) and securit\$3) and @ad<20000926) and ((fault or error or mistake or glitch\$2 or intrusion) with (predict\$3 or estimat\$3 or simulat\$3)) and (histor\$3 or past\$1)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/04 19:50
2	7	(((((FSM or IFSM or (finite adj state adj machine)) and simulat\$4) and network) and securit\$3) and @ad<20000926) and ((fault or error or mistake or glitch\$2 or intrusion) with (predict\$3 or estimat\$3 or simulat\$3)) and (histor\$3 or past\$1) and (random\$4 or statistic\$4 or stochastic\$2))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/04 19:59
3	7	(((((FSM or IFSM or (finite adj state adj machine)) and simulat\$4) and network) and securit\$3) and @ad<20000926) and ((fault or error or mistake or glitch\$2 or intrusion) with (predict\$3 or estimat\$3 or simulat\$3)) and (histor\$3 or past\$1) and (random\$4 or statistic\$4 or stochastic\$2)) and calculat\$4	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/04 20:03
4	7	((((((FSM or IFSM or (finite adj state adj machine)) and simulat\$4) and network) and securit\$3) and @ad<20000926) and ((fault or error or mistake or glitch\$2 or intrusion) with (predict\$3 or estimat\$3 or simulat\$3)) and (histor\$3 or past\$1) and (random\$4 or statistic\$4 or stochastic\$2)) and calculat\$4) and (function\$1 or equation\$1)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/04 20:47
5	1532	network\$1 and monitor\$4 and worm	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/04 20:14
6	15	(network\$1 and monitor\$4 and worm) and (fsm or (finite adj state adj machine))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/04 20:15
7	5	((network\$1 and monitor\$4 and worm) and (fsm or (finite adj state adj machine))) and @ad<20000926	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/04 20:15
8	10599	network\$1 and monitor\$4 and (virus\$2 or worm\$2 or intrusion\$1)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/04 20:15
9	85	(network\$1 and monitor\$4 and (virus\$2 or worm\$2 or intrusion\$1)) and (fsm or (finite adj state adj machine))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/04 20:15
10	50	((network\$1 and monitor\$4 and (virus\$2 or worm\$2 or intrusion\$1)) and (fsm or (finite adj state adj machine))) and @ad<20000926	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/04 20:15
11	15	((network\$1 and monitor\$4 and (virus\$2 or worm\$2 or intrusion\$1)) and (fsm or (finite adj state adj machine))) and @ad<20000926 and histor\$4	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/04 20:16
12	15	((network\$1 and monitor\$4 and (virus\$2 or worm\$2 or intrusion\$1)) and (fsm or (finite adj state adj machine))) and @ad<20000926 and histor\$4 and (random\$4 or statistic\$4 or stochastic\$2)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/04 20:16
14	4	(((((network\$1 and monitor\$4 and (virus\$2 or worm\$2 or intrusion\$1)) and (fsm or (finite adj state adj machine))) and @ad<20000926) and histor\$4) and (random\$4 or statistic\$4 or stochastic\$2)) and (random\$4 or stochastic\$2)) and equation\$1	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/04 20:17

13	9	(((((network\$1 and monitor\$4 and (virus\$2 or worm\$2 or intrusion\$1)) and (fsm or (finite adj state adj machine))) and @ad<20000926) and histor\$4) and (random\$4 or statistic\$4 or stochastic\$2)) and (random\$4 or stochastic\$2)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/04 20:18
15	6	("4985857" "5195095" "5295244" "5309448" "5317568" "5517622" "5528516" "5608720" "5661668" "5748896" "5799153" "5854750" "5918051" "5951680").pn. and histor\$4	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/04 20:54
16	2	((("4985857" "5195095" "5295244" "5309448" "5317568" "5517622" "5528516" "5608720" "5661668" "5748896" "5799153" "5854750" "5918051" "5951680").pn. and histor\$4) and equation\$1	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/04 20:54
-	1	(PISARSKY-VLADIMIR).in.	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/04 15:54
-	5	(PISARSKY).in.	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/04 16:22
-	5331	FSM or IFSM or (finite adj state adj machine)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/04 16:22
-	1182	(FSM or IFSM or (finite adj state adj machine)) and simulat\$4	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/04 16:23
-	689	((FSM or IFSM or (finite adj state adj machine)) and simulat\$4) and network	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/04 16:24
-	101	((((FSM or IFSM or (finite adj state adj machine)) and simulat\$4) and network) and securit\$3	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/04 16:24
-	57	(((((FSM or IFSM or (finite adj state adj machine)) and simulat\$4) and network) and securit\$3) and @ad<20000926	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/04 16:29
-	49	(((((FSM or IFSM or (finite adj state adj machine)) and simulat\$4) and network) and securit\$3) and @ad<20000926) and ((fault or error or mistake or glitch\$2 or intrusion) and (predict\$3 or estimat\$3 or simulat\$3)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/04 16:30
-	21	(((((FSM or IFSM or (finite adj state adj machine)) and simulat\$4) and network) and securit\$3) and @ad<20000926) and ((fault or error or mistake or glitch\$2 or intrusion) same (predict\$3 or estimat\$3 or simulat\$3))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/04 16:30
-	15	(((((FSM or IFSM or (finite adj state adj machine)) and simulat\$4) and network) and securit\$3) and @ad<20000926) and ((fault or error or mistake or glitch\$2 or intrusion) with (predict\$3 or estimat\$3 or simulat\$3))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/04 19:48



US Patent & Trademark Office

[Subscribe](#) (Full Service) [Register](#) (Limited Service, Free) [Login](#)

Search: The ACM Digital Library The Guide

((finite <near/1> state <near/1> machine) and monitor* and

SEARCH



[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used

finite near/1 state near/1 machine and monitor and simulat and network and securit and stochastic and control near/1 code and fault and histor and

Sort results by

publication date

Display results

condensed form

[Save results to a Binder](#)

[Search Tips](#)

[Open results in a new window](#)

[Try an Advanced Search](#)
[Try this search in The ACM Guide](#)

Results 81 - 100 of 200

Best 200 shown

Result page: [previous](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Relevance

81 [Information survivability control systems](#)

Kevin Sullivan, John C. Knight, Xing Du, Steve Geist

May 1999 **Proceedings of the 21st international conference on Software engineering**

Full text available: pdf(1.23 MB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

82 [Bimodal multicast](#)

Kenneth P. Birman, Mark Hayden, Oznur Ozkasap, Zhen Xiao, Mihai Budiu, Yaron Minsky

May 1999 **ACM Transactions on Computer Systems (TOCS)**, Volume 17 Issue 2

Full text available: pdf(302.06 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

83 [Automating parallel simulation using parallel time streams](#)

Victor Yau

April 1999 **ACM Transactions on Modeling and Computer Simulation (TOMACS)**, Volume 9 Issue 2

Full text available: pdf(194.69 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

84 [Summary of the sigmetrics symposium on parallel and distributed processing](#)

Jeffrey K. Hillingsworth, Barton P. Miller

March 1999 **ACM SIGMETRICS Performance Evaluation Review**, Volume 26 Issue 4

Full text available: pdf(1.17 MB)

Additional Information: [full citation](#), [index terms](#)



US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

Search: ☒ The ACM Digital Library ☐ The Guide

[Feedback](#) [Rep](#)

Terms used

finite near/1 state near/1 machine and monitor and simulat and network and securit and stochastic and cont

Sort results by

Display results

[Save results to a Binder](#)

[Search Tips](#)

☒ [Open results in a new window](#)

Try an
Try th

Results 61 - 80 of 200

Best 200 shown

Result page: [previous](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [n](#)

61 [Process migration](#)

September 2000 **ACM Computing Surveys (CSUR)**, Volume 32 Issue 3

Full text available: [pdf\(1.24 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [cit](#)

62 [Specification, validation, and synthesis of email agent controllers: A case study in function ric](#)

Robert J. Hall

August 2000 **Proceedings of the third workshop on Formal methods in software practice**

Full text available: [pdf\(527.90 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

63 [Real-time estimation of the parameters of long-range dependence](#)

Matthew Roughan, Darryl Veitch, Patrice Abry

August 2000 **IEEE/ACM Transactions on Networking (TON)**, Volume 8 Issue 4

Full text available: [pdf\(237.43 KB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

64 [Verisim: Formal analysis of network simulations](#)

Karthikeyan Bhargavan, Carl A. Gunter, Moonjoo Kim, Insup Lee, Davor Obradovic, Oleg Sokolsky, M

August 2000 **ACM SIGSOFT Software Engineering Notes , Proceedings of the International Analysis**, Volume 25 Issue 5

Full text available: [pdf\(325.27 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [cit](#)

65 [Binary translation and architecture convergence issues for IBM system/390](#)

Michael Gschwind, Kemal Ebcioglu, Erik Altman, Sumedh Sathaye

May 2000 **Proceedings of the 14th international conference on Supercomputing**

Full text available: [pdf\(1.44 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

66 [Software reliability and dependability: a roadmap](#)

Bev Littlewood, Lorenzo Strigini

May 2000 **Proceedings of the conference on The future of Software engineering**

Full text available: [pdf\(1.57 MB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

67 [Proactive computing](#)

David Tennenhouse

May 2000 **Communications of the ACM**, Volume 43 Issue 5

Full text available: [pdf\(270.77 KB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#), [review](#)

[html\(40.89 KB\)](#)

68 System-level power optimization: techniques and tools

Luca Benini, Giovanni de Micheli

April 2000 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume

Full text available: [pdf\(385.22 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

69 Session summaries from the 17th symposium on operating systems principle (SOSP'99)

Jay Lepreau, Eric Eide

April 2000 **ACM SIGOPS Operating Systems Review**, Volume 34 Issue 2

Full text available: [pdf\(3.15 MB\)](#)

Additional Information: [full citation](#), [index terms](#)

70 A brief history of cellular automata

Palash Sarkar

March 2000 **ACM Computing Surveys (CSUR)**, Volume 32 Issue 1

Full text available: [pdf\(283.32 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#), [review](#)

71 Programming languages and systems for prototyping concurrent applications

Wilhelm Hasselbring

March 2000 **ACM Computing Surveys (CSUR)**, Volume 32 Issue 1

Full text available: [pdf\(559.78 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#), [review](#)

72 Enforceable security policies

Fred B. Schneider

February 2000 **ACM Transactions on Information and System Security (TISSEC)**, Volume 3 Issue

Full text available: [pdf\(148.24 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [in](#)

73 Evolutionary design of complex software (EDCS) demonstration days 1999

Wayne Stidolph

January 2000 **ACM SIGSOFT Software Engineering Notes**, Volume 25 Issue 1

Full text available: [pdf\(1.90 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [index terms](#)

74 EROS: a fast capability system

Jonathan S. Shapiro, Jonathan M. Smith, David J. Farber

December 1999 **ACM SIGOPS Operating Systems Review , Proceedings of the seventeenth A principles**, Volume 33 Issue 5

Full text available: [pdf\(1.83 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [cit](#)

75 Strategic directions in simulation research (panel)

Ernest H. Page, David M. Nicol, Osman Balci, Richard M. Fujimoto, Paul A. Fishwick, Pierre L'Ecuyer,
December 1999 **Proceedings of the 31st conference on Winter simulation: Simulation---a bri**

Full text available: [pdf\(90.73 KB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index](#)

76 The CIP method: component- and model-based construction of embedded systems

Hugo Fierz

October 1999 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 7th Europea the 7th ACM SIGSOFT international symposium on Foundations of software**

Full text available: [pdf\(1.29 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [ind](#)

77 The holodeck ray cache: an interactive rendering system for global illumination in nondiffuse

Gregory Ward, Maryann Simmons

October 1999 **ACM Transactions on Graphics (TOG)**, Volume 18 Issue 4

Full text available: [pdf\(935.74 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index](#)

78 SASI enforcement of security policies: a retrospective

Úlfar Erlingsson, Fred B. Schneider

September 1999 **Proceedings of the 1999 workshop on New security paradigms**

Full text available:  [pdf\(862.14 KB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index](#)

79 Temporal sequence learning and data reduction for anomaly detection

Terran Lane, Carla E. Brodley

August 1999 **ACM Transactions on Information and System Security (TISSEC)**, Volume 2 Issue 3

Full text available:  [pdf\(628.31 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index term](#)

80 Inoculating software for survivability

Anup K. Ghosh, Jeffrey M. Voas

July 1999 **Communications of the ACM**, Volume 42 Issue 7

Full text available:  [pdf\(214.10 KB\)](#)  [html\(37.50 KB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Results 61 - 80 of 200

Result page: [previous](#) [1](#) [2](#) [3](#) **[4](#)** [5](#) [6](#) [7](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Play](#)



US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)Search: ☐ The ACM Digital Library ☐ The Guide**SEARCH**

The ACM Digital Library

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Simulating realistic network worm traffic for worm warning system design and testing

Full text [Pdf \(308 KB\)](#)

Source [Workshop On Rapid Malcode](#) [archive](#)
Proceedings of the 2003 ACM workshop on Rapid Malcode [table of contents](#)
Washington, DC, USA
SESSION: Network interactions [table of contents](#)
Pages: 24 - 33
Year of Publication: 2003
ISBN:1-58113-785-0

Authors [Michael Liljenstam](#) Dartmouth College, Hanover, NH
[David M. Nicol](#) Dartmouth College, Hanover, NH
[Vincent H. Berk](#) Dartmouth College, Hanover, NH
[Robert S. Gray](#) Dartmouth College, Hanover, NH

Sponsors [SIGSAC](#): ACM Special Interest Group on Security, Audit, and Control
[ACM](#): Association for Computing Machinery

Publisher ACM Press New York, NY, USA

Additional Information: [abstract](#) [references](#) [index terms](#) [collaborative colleagues](#)

Tools and Actions: [Discussions](#) [Find similar Articles](#) [Review this Article](#)
[Save this Article to a Binder](#) [Display in BibTex Format](#)

DOI Bookmark: - Use this link to bookmark this Article: <http://doi.acm.org/10.1145/948187.948193>
[What is a DOI?](#)

↑ ABSTRACT

Reproducing the effects of large-scale worm attacks in a laboratory setup in a realistic and reproducible manner is an important issue for the development of worm detection and defense systems. In this paper, we describe a worm simulation model we are developing to accurately model the large-scale spread dynamics of a worm and many aspects of its detailed effects on the network. We can model slow or fast worms with realistic scan rates on realistic IP address spaces and selectively model local detailed network behavior. We show how it can be used to generate realistic input traffic for a working prototype worm detection and tracking system, the Dartmouth ICMP BCC: System/Tracking and Fusion Engine (DIB:S/TRAFEN), allowing performance evaluation of the system under realistic conditions. Thus, we can answer important design questions relating to necessary detector coverage and noise filtering without deploying and operating a full system. Our experiments indicate that the tracking algorithms currently implemented in the DIB:S/TRAFEN system could detect attacks such as Code Red v2 and Sapphire/Slammer very early, even when monitoring a quite limited portion of the address space, but more sophisticated algorithms are being constructed to reduce the risk of false positives in the presence of significant "background noise" scanning.

↑ REFERENCES

Note: OCR errors may be found in this Reference List extracted from the full text article. ACM has

opted to expose the complete List rather than only correct and linked references.

- 1 Labrea. <http://www.hackbusters.net/LaBrea>.
- 2 Ssfnet web site. <http://www.ssfnet.org/>.
- 3 F. Baker. Rfc 1812: Requirements for IP version 4 routers. Request for Comments 1812, June 1995.
- 4 Vincent Berk, Wayne Chung, Valentino Crespi, George Cybenko, Robert Gray, Diego Hernando, Guofei Jiang, Han Li, and Yong Sheng. Process Query Systems for Surveillance and Awareness. In Proceedings of the SCI 2003, Orlando, Florida, July 2003.
- 5 Vincent H. Berk, Robert S. Gray, and George Bakos. Using Sensor Networks and Data Fusion for Early Detection of Active Worms. In Proceedings of AeroSense 2003: SPIE's 17th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls, Orlando, Florida, April 2003.
- 6 Z. Chen, L. Gao, and K. Kwiat. Modeling the Spread of Active Worms. INFOCOM 2003, 2003.
- 7 Brent N. Chun, Jason Lee, and Hakim Weatherspoon. Netbait: A distributed worm detection service. Available at <http://netbait.plain-lab.org/>, 2003.
- 8 Cisco. Dealing with mallocfail and high CPU utilization resulting from the "Code Red" worm. http://www.cisco.com/warp/public/-63/ts_codred_worm.shtml, October 2001.
- 9 J. Cowie, D. Nicol, and A. Ogielski. Modeling the Global Internet. IEEE Computing in Science and Engineering, 1(1):42--50, Jan.-Feb. 1999.
- 10 D.J. Daley and J. Gani. Epidemic Modelling: An Introduction. Cambridge University Press, Cambridge, UK, 1999.
- 11 Silicon Defense. Countermalice---Worm Containment System. <http://www.silicondefense.com/products/countermalice/>, 2003.
- 12 S. Floyd and V. Paxson. Difficulties in Simulating the Internet. IEEE/ACM Transactions on Networking, 9(4):392--403, August 2001.
- 13 J.O. Kephart and S.R. White. Measuring and Modeling Computer Virus Prevalence. Proceedings of the 1993 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, CA, May 1993.
- 14 M. Liljenstam, Y. Yuan, B.J. Premore, and D. Nicol. A Mixed Abstraction Level Model of Large-Scale Internet Worm Infestations. in Proc. of the Tenth IEEE/ACM Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), Fort Worth, TX, Oct 2002. IEEE Computer Society Press.
- 15 D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. Inside the Slammer Worm. IEEE Security and Privacy, 1(4):33--39, July 2003.
- 16 D. Moore, C. Shannon, and K. Claffy. Code-Red: A case study on the spread and victims of an Internet worm. in Proc. of the Internet Measurement Workshop (IMW), Marseille, France, Nov 2002. ACM Press.

- 17 David Moore, Colleen Shannon, Geoffrey M. Voelker, and Stefan Savage. Internet quarantine: Requirements for containing self-propagating code. In Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003), April 2003.
- 18 David Moore, Geoffrey M. Voelker, and Stefan Savage. Inferring Internet Denial-of-Service activity. In Proceedings of the 10th USENIX Security Symposium (USENIX'01), Washington, DC, August 2001.
- 19 Lawrence R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceeding of the IEEE, 77, Num. 2:257--286, 1989.
- 20 Donald B. Reid. An algorithm for Tracking Multiple Targets. IEEE Transactions on Automatic Control, AC-24, Num. 6:843--854, 1979.
- 21 S. Staniford. Code Red Analysis Pages: July infestation analysis. <http://www.silicondefense.com/cr/july.html>, 2001.
- 22 S. Staniford, V. Paxson, and N. Weaver. How to Own the Internet in Your Spare Time. in Proc. of the USENIX Security Symposium, 2002. <http://www.icir.org/vern/papers/cdc-usenix-sec02/index.html>.
- 23 A. Turner and M. Bing. project page (sourceforge). <http://tcpreplay.sourceforge.net/>, 2003.
- 24 I. van Beijnum. BGP. O'Reilly & Associates, Sebastopol, CA, 2002.
- 25 Vinod Yagneswaran, Paul Barford, and Johannes Ullrich. Internet intrusions: Global characteristics and prevalence. In Proceedings of the International Conference on Measurements and Modeling of Computer Systems (SIGMETRICS 2003), San Diego, California, June 2003.
- 26 C. Zou, L. Gao, W. Gong, and D. Towsley. Code Red Worm Propagation Modeling and Analysis. 9th ACM Conference on Computer and Communication Security (CCS), Washington DC, Nov 2002.
- 27 Cliff C. Zou, Lixin Gao, Weibo Gong, and Don Towsley. Monitoring and early warning for internet worms. Technical Report TR-CSE-03-01, University of Massachusetts at Amherst, 2003.

↑ INDEX TERMS

Primary Classification:

C. Computer Systems Organization

↳ C.4 PERFORMANCE OF SYSTEMS

↳ **Subjects:** Modeling techniques

Additional Classification:

C. Computer Systems Organization

↳ C.2 COMPUTER-COMMUNICATION NETWORKS

↳ C.2.0 General

↳ **Subjects:** Security and protection (e.g., firewalls)

↳ C.2.3 Network Operations

↳ **Subjects:** Network monitoring

↳ C.2.5 Local and Wide-Area Networks

↪ **Subjects:** Internet (e.g., TCP/IP)

General Terms:

Experimentation, Measurement, Performance, Security

Keywords:

code red, network modeling and simulation, network security, slammer, worm detection systems, worms

↑ **Collaborative Colleagues:**

<u>Robert S. Gray:</u>	<u>Joyce Barton</u>	<u>Peter Gerken</u>	<u>Saurab Nog</u>
	<u>Jeffrey Bradshaw</u>	<u>Arne Grimstrup</u>	<u>Ronald A. Peterson</u>
	<u>Maggie R. Breedy</u>	<u>James Hendler</u>	<u>Daniela Rus</u>
	<u>Marco Carvalho</u>	<u>Greg Hill</u>	<u>Niranjan Suri</u>
	<u>Daria A. Chacón</u>	<u>Gísli Hjálmtýsson</u>	<u>Kenneth R.</u>
	<u>Daria Chacon</u>	<u>Martin Hofmann</u>	<u>Whitebread</u>
	<u>Ezra E. K. Cooper</u>	<u>Martin O. Hofmann</u>	
	<u>Thomas Cowin</u>	<u>Renia Jeffers</u>	
	<u>George Cybenko</u>	<u>Guofei Jiang</u>	
	<u>Chris Garrett</u>	<u>David Kotz</u>	

Michael Liljenstam: Rassul Ayani
Jason Liu
Johan Montagnat
David M. Nicol
Andy T. Ogielski
L. Felipe Perrone
Robert Rönngren

<u>David M. Nicol:</u>	<u>Heidi R. Ammerlahn</u>	<u>Richard M. Fujimoto</u>	<u>Keith W. Miller</u>	<u>Subhas Roy</u>
	<u>Author:</u>	<u>Bruno Gaujal</u>	<u>Ravi Mirchandaney</u>	<u>Joel H. Saltz</u>
	<u>Osman Balci</u>	<u>Michael E. Goldsby</u>	<u>Larry J. Morell</u>	<u>Rahul Simha</u>
	<u>Shahid H. Bokhari</u>	<u>Albert G. Greenberg</u>	<u>Branson W. Murrill</u>	<u>Roger Smith</u>
	<u>Eric Carr</u>	<u>Philip Heidelberger</u>	<u>Vijay K. Naik</u>	<u>Roger M. Smith</u>
	<u>Gianfranco Ciardo</u>	<u>Robert R. Henry</u>	<u>Robert E. Noonan</u>	<u>Jeffrey S. Steinman</u>
	<u>Gianfranco F. Ciardo</u>	<u>Michael M. Johnson</u>	<u>David R. O'Hallaron</u>	<u>King Tan</u>
	<u>Dartmouth College</u>	<u>Simon H. Kahan</u>	<u>Ernest H. Page</u>	<u>James C. Townsend</u>
	<u>Thomas H. Cormen</u>	<u>Pierre L'Ecuyer</u>	<u>Daniel L. Palumbo</u>	<u>Don Towsley</u>
	<u>James H. Cowie</u>	<u>Craig A. Lee</u>	<u>Stephen K. Park</u>	<u>Kishor S. Trivedi</u>
	<u>Thomas W. Crockett</u>	<u>Scott Leutenegger</u>	<u>L. Felipe Perrone</u>	<u>Michael Ulrey</u>
	<u>Kay Crowley</u>	<u>Scott T. Leutenegger</u>	<u>Luiz Felipe De Lima</u>	<u>Jeffrey M. Voas</u>
	<u>Mike Devetsikiotis</u>	<u>Michael Liljenstam</u>	<u>Perrone</u>	<u>Jake Wegmann</u>
	<u>Phillip M. Dickens</u>	<u>Jason Liu</u>	<u>Anna L. Poplawski</u>	<u>Frank H. Willard</u>
	<u>J. M. Duva</u>	<u>Xiaowen Liu</u>	<u>Brian J. Premore</u>	<u>Linda F. Wilson</u>
	<u>John Mark Duva</u>	<u>Malcolm Yoke Hean</u>	<u>Paul F. Reynolds</u>	<u>Paul E. Wright</u>
	<u>and J.M. Duva</u>	<u>Low</u>	<u>Paul F. Reynolds</u>	<u>Ann S. Yoshimura</u>
	<u>Paul A. Fishwick</u>	<u>Boris D. Lubachevsky</u>	<u>Scott E. Riffe</u>	
		<u>Weizhen Mao</u>		

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2003 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)

Simulating Realistic Network Worm Traffic for Worm Warning System Design and Testing

Michael Liljenstam David M. Nicol Vincent H. Berk Robert S. Gray

{mili,nicol,vberk,rgray}@ists.dartmouth.edu
Institute for Security Technology Studies
Dartmouth College
45 Lyme Rd., Suite 300
Hanover, NH 03755

ABSTRACT

Reproducing the effects of large-scale worm attacks in a laboratory setup in a realistic and reproducible manner is an important issue for the development of worm detection and defense systems. In this paper, we describe a worm simulation model we are developing to accurately model the large-scale spread dynamics of a worm and many aspects of its detailed effects on the network. We can model slow or fast worms with realistic scan rates on realistic IP address spaces and selectively model local detailed network behavior. We show how it can be used to generate realistic input traffic for a working prototype worm detection and tracking system, the Dartmouth ICMP BCC: System/Tracking and Fusion Engine (DIB:S/TRAFEN), allowing performance evaluation of the system under realistic conditions. Thus, we can answer important design questions relating to necessary detector coverage and noise filtering without deploying and operating a full system. Our experiments indicate that the tracking algorithms currently implemented in the DIB:S/TRAFEN system could detect attacks such as Code Red v2 and Sapphire/Slammer very early, even when monitoring a quite limited portion of the address space, but more sophisticated algorithms are being constructed to reduce the risk of false positives in the presence of significant “background noise” scanning.

Categories and Subject Descriptors

C.4 [Performance of Systems]: Modeling techniques—simulation; C.2.0 [Computer and Communication Networks]: Security and Protection—worms; C.2.3 [Network Operations]: Network monitoring—worm detection; C.2.5 [Local and Wide-Area Networks]: Internet

General Terms

Security, Experimentation, Measurement, Performance

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WORM’03, October 27, 2003, Washington, DC, USA.

Copyright 2003 ACM 1-58113-785-0/03/0010 ...\$5.00.

Keywords

Network Security, Network Modeling and Simulation, Worms, Worm Detection Systems, Code Red, Slammer

1. INTRODUCTION

Network worm infestations over the last few years, such as Code Red [16, 22], Code Red II [16, 22], and Nimda [22] in 2001, and recently the Sapphire/Slammer [15] worm have focused a lot of attention on threats posed by self-replicating malicious code. As a result, efforts are under way to create early warning systems and various defense mechanisms, e.g. [5, 26, 11, 1], and the general feasibility of such efforts is the focus of studies such as [17, 6].

However, since large-scale worm events are fortunately still rare, it is not possible to test prototype designs in a live setting. Moreover, faithfully reproducing the effects of large-scale worm attacks in a laboratory setup is difficult. In this paper we describe a worm simulation model we are developing, and how it is being used to generate test data for the Dartmouth DIB:S/TRAFEN System [5]. This particular worm detection system collects copies of ICMP messages generated by random scanning and tries to recognize signatures of early worm propagation by correlating the collected messages. Questions regarding things like necessary collection point coverage, “signal-to-noise” ratio for reliable detection, and optimal parameter settings in the correlation algorithms are all difficult to answer without some way to generate realistic test traffic.

Accurately simulating the effects that large-scale worm infestations have on infrastructure is challenging since, in addition to issues related to heterogeneity and change in the Internet [12]: *i*) a worm that infects tens or hundreds of thousands of machines on the Internet gives rise to an *inherently large-scale phenomenon*, and requires the model to be of appropriate scale to correctly model the propagation dynamics; *ii*) with few exceptions, most worms have propagated over time scales of hours to days (or longer), thus it may result in a *large span of timescales* where network events at timescales down to microseconds are simulated over days. This could be further complicated in the case of “stealthy worms” propagating slowly to avoid detection.

For a credible model we want the worm spreading through a network with a *realistic number of hosts* using *realistic scan*

rates over a realistic address space. This will ensure that the propagation dynamics are realistic and will avoid artifacts as we extract more detailed information from the model. The model we describe combines modeling at multiple levels of abstraction in order to be both detailed enough to generate realistic packet traffic, and efficient enough to model a worm spreading through the Internet. We are developing this model with the aim to provide a general testbed for worm detection and countermeasure systems. It is currently under active development and is being publicly released as an add on package (called *SSF.App.Worm*) to the SSFNet simulator [2][9].

The remainder of this paper is organized as follows: We introduce the DIB:S/TRAFEN system in Section 2, the simulation model in Section 3–4, and validate the model in Section 5. Section 6 describes our case study, where we use the simulator to test the detection algorithms currently implemented DIB:S/TRAFEN. We discuss related work in Section 7, and finally conclude in Section 8.

2. DIB:S/TRAFEN OVERVIEW

The focus of this paper is the worm simulation model, but we start by introducing the DIB:S/TRAFEN system [5], which sets the scene for the simulation and the case study presented later. DIB:S/TRAFEN can detect and classify active Internet worms in their earliest stages of propagation, increasing the chance of intervention and subsequent mitigation of Internet-scale epidemics.

Most current active worms spread by randomly probing IP addresses and, since the IPv4 address space is densely populated, even unbiased random scanning will find vulnerable systems relatively quickly. This random scanning however, will probe many unassigned IP addresses, i.e., addresses that are not associated with a reachable computer. In many cases, routers that receive a packet destined for an unreachable IP address will drop the packet and return an *ICMP Destination Unreachable* (ICMP Type 3) message to the packet originator. This ICMP-T3 message will include the original IP header and at least 8 bytes of the protocol header, which together will include the source and destination IP addresses and port numbers for both UDP and TCP packets. This embedded data makes ICMP-T3 messages very useful for detection of scanning events, and, in fact, our DIB:S system collects these messages by having a select group of participating routers forward all the ICMP-T3 messages that they generate to an analysis station. The generation of ICMP-T3 messages is rate limited, usually at 3 per second, and will add negligible overhead to the router.¹ In addition, if site policy so dictates, the ICMP-T3 message can be sent to the analysis station, but not sent to the scanning machine, preventing easy reconnaissance of the network. Finally, although we will see later that the total number of participating routers can be small, these routers must be distributed across a significant fraction of the Internet address space to ensure timely and accurate worm detection.

¹Note that routers usually apply this rate limit to all outgoing ICMP messages, rather than to individual ICMP message subtypes or network interfaces. Traffic, on any interface, that generates ICMP messages will reduce the number of worm-related ICMP-T3 messages sent to the analysis station. Initial experience indicates that this bias has only a modest effect across a group of instrumented routers.

As the ICMP-T3 messages arrive at the DIB:S analysis station, they are sorted and analyzed according to the embedded source and destination addresses and ports. When the total number of packets for any source or destination machine or port exceeds a threshold N_{DIBS} within a configurable time interval Δt_{DIBS} , DIB:S generates a scan alert, and DIB:S will not generate another scan alert for the same machine and port combination until Δt_{DIBS} seconds elapse. DIB:S generates several types of scan alert, corresponding to a single source machine, multiple source machines, and so on, but the most important scan alert for worm detection is when a single source machine uses the same protocol P to contact the same port p on N_{DIBS} target machines within Δt_{DIBS} seconds. This alert indicates the “bloom” typical of random scanning behavior originating from a single host, and an exponential increase in the number of alerts for the same port and protocol most likely indicates a propagating worm.

It is the job of TRAFEN to detect this exponential increase. TRAFEN, which stands for TRacking And Fusion ENgine and is an implementation of a *Process Query System* [4], is a domain-independent middleware that allows the rapid development of tracking and data-fusion applications. The developer defines a process model that describes how to identify which incoming observations correspond to the same real-world process, such as a propagating worm or a vehicle moving through a battlespace, and how to predict the future state of a previously identified process. The process model can be defined in many ways, such as Hidden Markov Models [19], Kalman filters, or domain-specific rule-bases. TRAFEN uses traditional tracking algorithms, most notably an implementation of Reid’s multiple hypothesis tracking (MHT) algorithm [20], to handle the mechanics of constructing the most likely observation groupings. Specifically, TRAFEN keeps multiple *hypotheses*, where each hypothesis is a set of *tracks* of related observations. Each observation is represented only once in each hypothesis, and each hypothesis aims to represent an accurate view of the world. Using the process model, each incoming observation is compared with each track in each hypothesis, and one or more new hypotheses, each containing the new observation, are generated for each existing hypothesis. The hypotheses are ranked according to their likelihood, and then are pruned to prevent an exponential increase as further observations arrive. In our case, the observations are DIB:S alerts, and our current working prototype uses a simple rule-based process model to identify worm activity.

Collecting ICMP-T3 messages from instrumented routers, rather than using ingress and egress filters at network boundaries, allows efficient detection of scans that span multiple networks, even if the scan probes only one address from each network. Moreover, using instrumented routers associated with populated address space, rather than only with entirely unallocated address blocks, makes it harder for worm authors to avoid the detection system. DIB:S and TRAFEN could use multiple types of scan data to improve their detection performance, however.

3. THE WORM SIMULATION

For the worm simulation, our starting point is the SSFNet simulator [2][9], a packet-level network simulator written in Java that supports parallel and distributed execution for increased scalability. It includes standard TCP/IP protocols

(IP, ICMP, TCP, UDP, HTTP, ...) and detailed implementations of routing protocols such as BGP and OSPFv2.

The challenge of simulating worm events involving as many as hundreds of thousands of hosts generating high rates of scan packets can quickly lead to resource and performance demands that are beyond even state of the art packet-level simulators running on super-computers. Consider, for instance, the Slammer worm [15], which infected at least 75,000 hosts with an estimated average scan rate of about 4000 scans/s per worm.² Then, if we assume this scan rate at the peak of infection, we would have to simulate $75,000 \cdot 4000 \cdot h = 3 \cdot 10^8 \cdot h$ Packet Transmission Events per second of simulated time, where h is the mean number of hops a scan packet travels. This, and the memory and effort required to model significant portions of the Internet at this level of detail, has led us to explore a simulation that mixes modeling at dual levels of abstraction.

The propagation dynamics of worms spreading by uniform random scanning, such as Code Red, lend themselves well to a coarse form of modeling based on epidemic models. This drastically reduces resource requirements compared to a packet-level model of the whole system. On the other hand, epidemic models alone include no information about the underlying network and its possible effects on the propagation, e.g., bandwidth constraints on scans as observed during Slammer or router failures observed due to worm traffic [24, 8]. Moreover, additional details are required to create meaningful, realistic test data for detection systems. Consequently, we have chosen to combine a coarse-level model of the worm propagation and scan traffic with detailed models (packet-level simulation) of selected parts of the network. More details on the benefits achieved and trade-offs involved in this approach can be found in [14] where we proposed the mixed abstraction-level model.

Figure 1 illustrates how a coarse “macroscopic model”, such as an epidemic model of the worm spreading, drives the network model in terms of host infections and scan traffic induced in the network. In the simplest case, we assume that the worm spread is not affected by what goes on at the network level, although other “flavors” of macroscopic models also allow us to take network effects (such as failures, routing dynamics, or congestion) into account.

3.1 Homogeneous Deterministic Epidemic

The simplest form of macroscopic model implemented in the simulator is the continuous-time deterministic version of the “general epidemic model” [10]. The model assumes a fixed population of size N and describes the evolution of the system through a set of equations:

$$\frac{ds(t)}{dt} = -\beta s(t)i(t) \quad (1)$$

$$\frac{di(t)}{dt} = \beta s(t)i(t) - \gamma i(t) \quad (2)$$

$$\frac{dr(t)}{dt} = \gamma i(t) \quad (3)$$

where the constant β is the *infection parameter*, i.e., the pairwise rate of infection, and the constant γ is the *removal parameter*. These equations describe the rate of transitions from the population of susceptible hosts s to the

²This scan rate was observed early, before bandwidth limitations set in. Thus, at the peak of infection, it should have been somewhat lower.

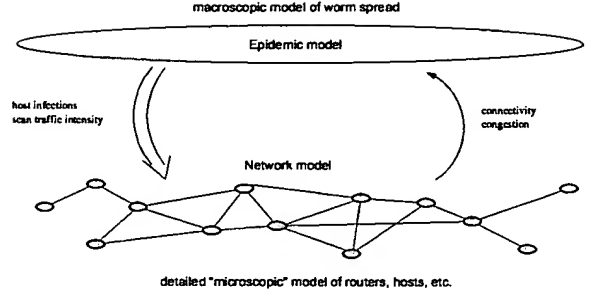


Figure 1: The simulation uses models at two different levels of abstraction: a significant fraction of the Internet is modeled coarsely at the “macroscopic” level, and selected parts are modeled using full packet level simulation. The coarse model of the worm infection drives the network model, and some (limited) feedback may occur, such as accounting for changes in connectivity.

infected population i to the removed population r . Thus, $s(t) + i(t) + r(t) = N, \forall t \geq 0$. Here we assume that the total population under consideration is large enough so that mean evolution of the stochastic system can be approximated using this “law of mass action”-formulation, and we assume that all (susceptible) hosts in the Internet can be viewed as one homogeneous population. Furthermore, it assumes *homogeneous mixing*—meaning roughly that interaction is equally likely between all members of the population in a given (small) interval of time.

A similar model appears to first have been proposed for the Code Red worm in [21], and epidemic models have since been considered in several studies [22, 14, 16, 26, 17].

3.2 Stochastic Epidemic (Homogeneous)

For smaller populations or early stages of propagation, it is important to model the stochastic evolution of the system for better fidelity. Consider a worm spreading by sending scans (infection packets) at random with uniform distribution over the 32 bit IPv4 address space. Thus, a scan from an infected host will hit a specific address with probability $P[\text{hit}] = 2^{-32}$.³ If σ is the scan rate (to unique addresses) of a single worm, then i infected hosts generate $\sigma \cdot i$ scans. Thus, if s is the number of susceptible hosts, then the number of new infections X in a time interval Δt has a Binomial distribution with parameters $\text{Bin}(s, p)$ where $p = P[\text{hit}] \cdot i \cdot \sigma \cdot \Delta t$. If X_t is the number of infections at time t and Y_t is the number of removals (similarly defined), then a discrete time model of the system can be written as:

$$s(t + \Delta t) = s(t) - X_t \quad (4)$$

$$i(t + \Delta t) = i(t) + X_t - Y_t \quad (5)$$

$$r(t + \Delta t) = r(t) + Y_t \quad (6)$$

Since X has a Binomial distribution, $E[X] = s \cdot p = s \cdot P[\text{Hit}] \cdot i \cdot \sigma \cdot \Delta t = \{\text{regrouping}\} = (P[\text{Hit}] \cdot \sigma \cdot \Delta t) \cdot s \cdot i$. Comparing back to Equation 1, it is easy to see how

³This is slightly simplified as it ignores reserved address space for multicast and loopback, but taking these address blocks into account makes no significant difference.

the deterministic model describes the mean evolution of the system and that $\beta = \sigma \cdot \Delta t \cdot P[Hit]$. Note also that it is the uniform distribution of the scans that ensures that the homogeneous mixing assumption is true. For instance, it does not hold for viruses propagated by emails or magnetic media since these interactions are typically constrained by human social relationships [13].

3.3 Spatial Epidemic

To study scan traffic flows, we break the model down spatially by using the ‘stratified population’ formulation of the epidemic model [10], where (in this case) the host population is stratified by network:

$$\begin{aligned}\frac{ds_j(t)}{dt} &= -s_j(t) \cdot (\beta_{1j}i_1(t) + \dots + \beta_{mj}i_m(t)) \\ \frac{di_j(t)}{dt} &= s_j(t) \cdot (\beta_{1j}i_1(t) + \dots + \beta_{mj}i_m(t)) - \gamma_j i_j(t) \\ \frac{dr_j(t)}{dt} &= \gamma_j i_j(t)\end{aligned}$$

where, for each population group j : $s_j(t)$, $i_j(t)$, and $r_j(t)$ are the state variables, γ_j is the removal parameter, and β_{ij} is the infection parameter between groups i and j .

For a system that behaves as the homogeneous model, we set $\beta_{ij} = \beta$, since this gives us (ignoring removals for clarity)

$$\sum_j \frac{di_j(t)}{dt} = \beta \cdot i(t) \cdot \sum_j s_j(t) = \beta \cdot s(t) \cdot i(t)$$

i.e., the same infection growth rate as the homogeneous case. A corresponding stochastic model can also easily be created, as in the previous section.

3.4 Scan Traffic

Given a spatial model of the spread, we can easily calculate the mean traffic intensity for egress and ingress scans to each network. We have scans generated from network j at rate $\sigma_j^{gen}(t) = i_j(t) \cdot \sigma$ and scans destined for network j at rate $\sigma_j^{dest}(t) = i(t) \cdot \sigma \cdot \frac{A_j}{2^{32}}$, where A_j is the size of network j ’s address space. Hence, for network j we have egress scans $\sigma_j^{egr}(t) = \sigma_j^{gen}(t) \cdot (1 - \frac{A_j}{2^{32}})$ and ingress scans $\sigma_j^{igr}(t) = \sigma_j^{dest}(t) - \sigma_j^{gen}(t) \cdot \frac{A_j}{2^{32}}$. This simple model is sufficient for studying scan traffic passing through gateway routers of ‘edge’ networks if we assume networks with a single gateway. Moreover, if we assume that we study ASes and let each network in our model be an AS, the model is simple enough to allow us to simulate the entire AS-topology of the Internet.

Routers covering ‘edge networks’ are expected to be useful detectors as they will generate ICMPs for unreachable hosts. Core backbone routers could be useful for covering more network traffic, but on the other hand can only ‘detect’ traffic going to bogus (unadvertised) prefixes. Thus, in the following, we will model the effect of equipping edge networks gateway routers with ICMP BCC: capabilities for the DIB:S/TRAFEN system. Note that this means we do not need to model the traffic flows between ASes or routers ‘internally’ in the AS topology. Hence, the simulator loads an AS-topology model, based on adjacencies from a BGP table dump. We stress, however, that it is only the size of the networks that is relevant here.⁴

⁴We thus avoid issues of incompleteness of the AS-graph and routing policies that constrain possible traffic flows.

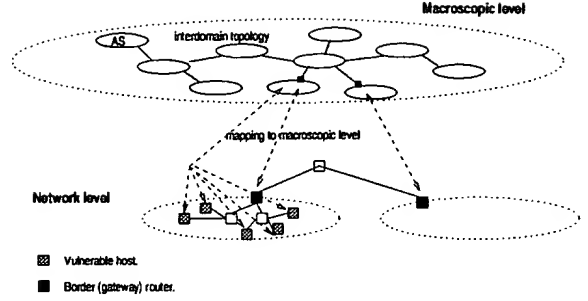


Figure 2: Certain key entities in the detailed network model, such as vulnerable hosts and gateway routers, can be mapped to corresponding entities at the macroscopic level.

The worm-modeling framework allows the user to map certain entities in the network model, i.e., elements represented at full packet-level detail using standard SSFNet constructs, to corresponding entities in the macroscopic model. Thus, it is possible to model the whole Internet (or a large part of it) coarsely, and selected parts in more detail, as illustrated in Figure 2. We return later to how this modeling pattern is used to generate packet-level information from the macroscopic model.

In this study, we simulate two observed worms, Code Red v2 (CRv2) on July 19, 2001 [16, 22] and Sapphire/Slammer (Slammer) on Jan 25, 2003 [15], to demonstrate the validity of the model. It is also possible to generalize over the parameter space of plausible hypothetical worms, however. The parameter choices and other specifics for the CRv2 and Slammer worm simulations will be described later in Section 5 where we also validate these models by comparing them to real captured traffic during the attacks.

4. GENERATING TEST TRAFFIC

As illustrated in Figure 3 the DIB:S/TRAFEN system operates by collecting copies of ICMP type 3 (unreachable) messages. Instrumented routers in the Internet send copies of ICMP type 3 messages to the DIB:S system which correlates and analyzes the data.

We simulate AS networks at the macroscopic level, that is, abstracting away internal details. A subset of the ASes are assumed to have a single gateway router which is instrumented to send ICMP copies. These instrumented routers are modeled at the detailed (packet) level and mapped to the macroscopic level as was shown in Figure 2. The DIB:S collection point host is also modeled at the detailed (packet) level, residing in a separate AS.

In this study we focus on ingress scans observed at the instrumented gateway routers and do not generate ICMPs for any egress scans that might be coming from within the router’s network. The instrumented router model has to translate the observed ingress scan rate into a packet process and generate ICMP packets corresponding to observed scans that target an address where there is no host. As shown in Figure 3, all packets arriving to the DIB:S system in the model are dumped to a file in binary tcpdump format. Using the tcpreplay tool [23], we can then replay the packet stream into the real DIB:S system to simulate

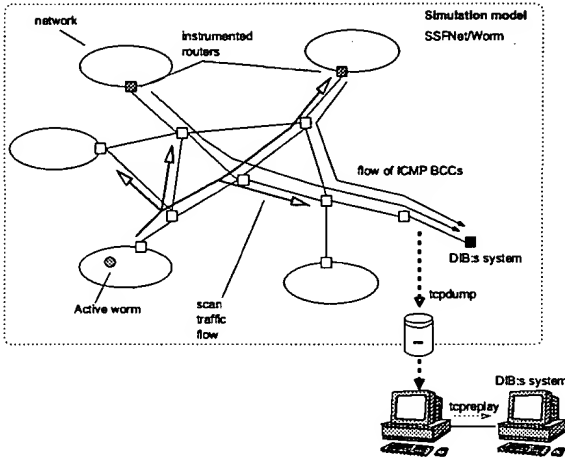


Figure 3: Simulation of worm and DIB:S system. Instrumented routers send “Blind Carbon Copies” of generated type 3 ICMP messages to the DIB:S system. The system analyzes the ICMPs to identify scanning activity, and correlates that scanning activity to track worm infection. The worm and collection system are simulated, and the ICMP copies arriving at the DIB:S system are dumped to file and then replayed against the real DIB:S system host.

the ICMP packets observed during the attack. However, DIB:S is also able to directly read tcpdump files for running experiments faster than real time.

From Flows to Packets. The ingress scan flows are modeled (at the macroscopic level) as a piecewise constant flow rate. At the instrumented routers, these flows must be converted into a packet arrival process, and because we are adding information, any such conversion will, by necessity, be an approximation. Since the incoming scans at a router are generated independently by many sources, we have chosen (based solely on this argument) to model the arrivals of scans (for distinct destination IP addresses) as a Poisson process. At the beginning of each time-step interval the arrival rate is set to the ingress scan flow rate $\sigma_j^{ingr}(t)$. Inter-arrival times are sampled and the next packet arrival determined. However, if the next packet arrival falls beyond of the current time-step it is discarded and a new sample taken at the beginning of the next time-step. This prevents prolonged quiet periods as the scan rates gradually grow from very low levels and will accurately model a Poisson process once $\sigma_j^{ingr}(t) \cdot \Delta t \gg 1$.

If the worm uses a transport protocol that generates retry packets, e.g. CRv2 where TCP generates SYN retries, those packets are added. Captured CRv2 packet traces indicate that the Microsoft TCP implementation generates a retry 3 seconds after the initial SYN, and a second retry 6 seconds after the first retry. These retries are scheduled accordingly into the packet arrival process when TCP is simulated.

For each network j a host address utilization fraction u_j is configured, denoting the fraction of addresses occupied by hosts. A Bernoulli trial process with probability $1 - u_j$ determines if an ICMP should be generated for an arriving scan packet.

Finally, RFC 1812 [3] states that routers should be able to rate limit the generation of ICMP messages to reduce load. We have observed that this is commonly used, with typical rate limits in the range of 3–4 ICMPs per second. A rate limit of 3 ICMPs/s was implemented in the simulator and was used in the experiments.

Generating ICMP Packets. Certain ICMP packet content is used by the DIB:S system correlation, and thus has to be generated with some care:

IP header SSFNet generates the IP header. The simulated source and destination addresses are replaced by real source and destination addresses as the packets are dumped to the packet trace file. This way the convenience of SSFNet’s automatic IP address assignment is not sacrificed.

ICMP header We let ingress scans result in type 3 subtype 1 “host unreachable” packets.

Embedded headers (IP/TCP/UDP) Each host infected by the worm is assigned a unique IP address sampled uniformly over the address space assigned to the network in which the host resides. The source IP address is drawn with equal probability among all active worm copies. The destination IP address is drawn uniformly from the destination network address space. The destination port number is known for the worm in question, and the source port is drawn with uniform distribution from ports above the “well-known” range, i.e., from 1024–65535.

Background Noise. It is useful to also be able to generate “background scanning noise” for at least two reasons: *i)* to model cases, such as for the CRv2 worm, where there was a significant level of scans going on before the worm was launched, and *ii)* to test what “signal-to-noise ratio” is necessary for reliable detection. The scans before CRv2 were partly due to version 1 of the worm (that was ineffective in spreading) and partly because TCP port 80 (HTTP) is simply a popular port to probe for vulnerabilities.

We have included the option of adding background noise with a fixed rate of scans in the simulator. Again, since we expect that the scans are generated by a large number of independent sources, we model the arrivals using a Poisson process, which is simply superimposed on other scan arrivals. For background scans, the source IP addresses are expected to lack time locality, so they are sampled uniformly over the whole IPv4 address space.

5. MODEL PARAMETERS AND VALIDATION

We validate our simulation by comparing output from the model with real captured scan traces from the Code Red v2 and Slammer attacks. The simulation model used for this comparison is the spatially distributed deterministic model where β is calculated from the scan rate. It is more convenient to use the deterministic version of the model for comparisons with real data since the infection ‘takes off’ at a predictable time point. However, *for the actual testing, we use the stochastic model* as it captures the important variability inherent in propagation from small infected populations.

Code Red v2. Using scan traffic data collected by the Chemical Abstract Service (CAS) during the Code Red v2 attack, we estimate the number of infected hosts over time using the bias adjustment method proposed by Zou et al. [27]. Figure 4 shows counts of unique source IP addresses in scans

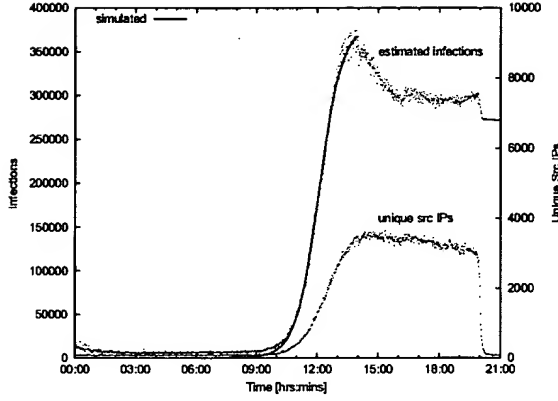


Figure 4: Raw counts of unique scan source IPs captured at CAS for Code Red v2 on July 19th, 2001, in 1 minute bins are shown (with respect to the right-hand axis). Also shown is a comparison of estimated infections based on the source IP counts (dots) and infections in the simulation (line).

destined for the CAS /16 network on TCP port 80 (on the right hand side y-axis), and it shows the estimated global number of infected hosts after bias adjustment (on the left hand axis).⁵ Since our primary interest is the initial growth of infection, we do not model host patching or filtering and set $\gamma = 0$, and, as the attack occurred over the span of several hours we use a fairly coarse time step $\Delta t = 60$ seconds.

These estimates indicate at least 360,000 susceptible hosts (which agrees well with [16]), and running Code Red v2 in a lab setup we have observed a mean scan rate slightly higher than 5 scans per second. Based on the estimated infections, we set the population of susceptibles $N = 380,000$ and the scan rate $\sigma = 5.65$ scans/s, since these produce a reasonably good fit to the growth phase of the infection as shown with the solid line in Figure 4.

It can be noted that whereas the simulation starts off with a single infection, the estimates start at a higher level. As was mentioned in Section 4, the CRv2 attack was preceded by a significant level of background noise, in part due to an earlier version of Code Red (v1) that was released earlier, but was less effective in spreading.

As a validation step, we pick a /16 network in the model (same size as the CAS network) and count all incoming scan SYN packets, just like the raw data for the real CAS network. The result is shown in Figure 5 and reveals a puzzling aspect of the real captured data set: at the peak of infection, the real data contains an almost 50% higher scan packet rate than would be expected.⁶

⁵The bias adjustment essentially attempts to infer the true number of active worms based on the cumulative count and difference in numbers of unique scan source IPs observed on a limited address space.

⁶A simple calculation shows that the simulator produces the expected result given the assumptions: At the peak of infection, i is close to 380,000, and $\sigma = 5.65$ scans/s. Let $E[X]$ be the expected number of scan packets observed during a one minute interval. Then $X \in \text{Bin}(n, p)$ with $n = i \cdot 3\sigma$ and $p = 2^{-(32-16)}$. Thus, at the peak of infection,

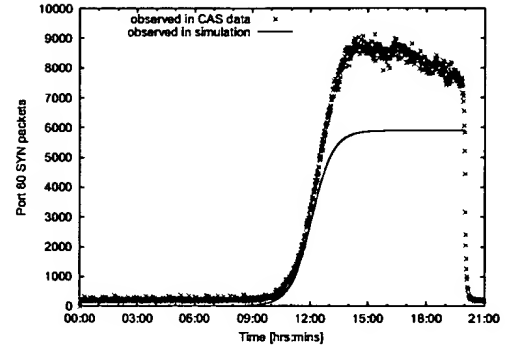


Figure 5: Comparison of total number of SYN packets captured in one minute bins on the CAS network during CRv2 with observations on a /16 network in the simulation.

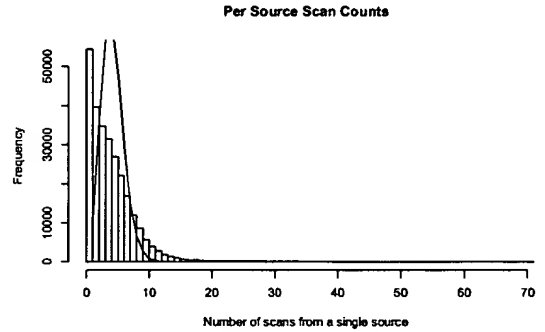


Figure 6: Histogram of the number of scans (excluding SYN retries) from each source IP within the 10 hour attack period compared to the expected Binomial distribution.

Essentially, there is a discrepancy between the observed number of unique source addresses in the scans and the total number of scans in the captured data. To understand this better, we examined the packet trace and counted the number of scans received from each source IP address. Figure 6 shows a histogram of the scans per source count. During the 10 hour attack period, the trace contains 269,695 unique sources sending 1,325,753 SYNs in total. The graph also shows a Binomial distribution where it is assumed that all sources were active for the entire period. Since many sources started scanning at some later point, there will tend to be more low counts than predicted by the distribution, as can be seen in the graph. However, the histogram also shows more mass in the tail than expected, i.e., many sources sending an unexpectedly high number of scans. In fact, whereas the expected number of scans from a single source in this time period is approximately 3.1, one source sent as many as 251 scans to this network.

Based on this, we conclude that the trace contains a higher scan rate per source than expected from scan rate exper-

$$E[X] = n \cdot p \approx 5900 \text{ scans/min.}$$

inents and a uniform scan distribution. There could be many possible explanations for this, from imperfections in the worm's random number generation to possibilities of Network Address Translation gateways or reinfections. Ultimately, this discrepancy will not affect our tests significantly, as we are testing detection during the early stages of growth where the model corresponds well with reality. Thus, we prefer to use this model, which is well understood, rather than adding any ad-hoc terms to account for the difference.

Sapphire/Slammer. We base our model of the Slammer worm largely on the analysis in [15] and a data set we have obtained from TRIUMF Canada. This worm spread much faster than CRv2 and is estimated to have infected most of the vulnerable hosts within 10 minutes. Modeling this worm also involves a few more complications: Firstly, the scan rate was essentially *bandwidth constrained* which affected the propagation speed and possibly also the observations on a single network (i.e., the incoming scan traffic that could be observed is also likely to be limited at some point). Secondly, code analysis in [15] indicated that it had faulty random number generator code. Thus, some address spaces would never be scanned by a single worm, although a large number of worms are likely to jointly cover the entire address space. The net effect is that collected data sets from any single site must be viewed with some caution, especially for a small number of worms. We have included facilities to model access-link bandwidth limitations in the simulator, but found that the worm detection point (the feature we want to test in this study) occurs so early on that the bandwidth constraints can be ignored in the model. We model the worm scanning using uniform distribution over the entire IPv4 space (not taking the defects into account) and thus effectively assume that any measurements taken in the model are from networks that would not have been bypassed by the worm scans. Because of the speed of the worm, we use a smaller time step $\Delta t = 1$ second, and again since our primary interest is in the initial stages of worm spread, we set $\gamma = 0$ (no patching or filtering).

Using TRIUMF Canada data, we estimated the number of infected hosts as the Slammer worm spread. We expect to see exponential growth in the early stages before bandwidth limitations set in. Figure 7, top graph, shows a semilog plot of the first two minutes of the worm data where there appears to be a short straight line segment, indicating exponential growth. [15] states that by observing data from multiple networks, they found at least 75,000 infected hosts, an average scan rate $\sigma \approx 4000$ scans/s, and 7 ± 1 new infections/min by a single worm during early spread. At startup from a single infection, this means $i(0) = 1$ and $\frac{di(t)}{dt}|_{t=0} = \beta s(0)i(0) = \beta s(0) = 7 \pm 1$ infections per minute, with $\beta = \sigma \cdot \Delta t \cdot P[Hit] \approx 5.6 \cdot 10^{-5}$. This would mean $s(0)$ in the range of 107,000–143,000. We set $N = 120,000$ resulting in a fairly good fit for the first two minutes of propagation, as also shown in the top graph of Figure 7. The bottom graph in Figure 7 shows the same plot with linear axes where it is more evident how the infection growth rate slows down compared to the model. However, the early exponential growth stage is the time period that is relevant for the tests performed in this study and during that stage the model matches reality well.

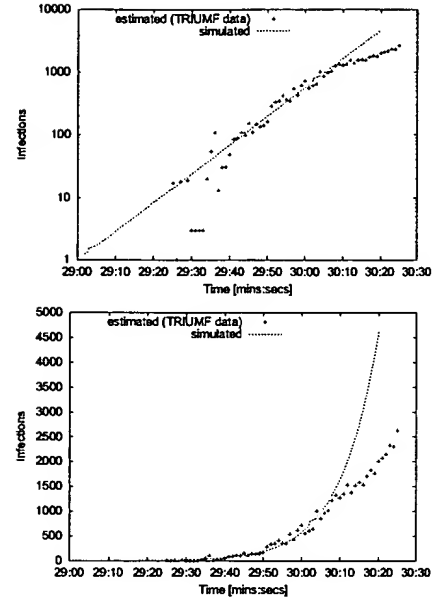


Figure 7: The first couple of minutes of the Sapphire/Slammer attack as captured at TRIUMF Canada. The graphs show estimated number of infections (calculated from unique source IPs) compared to a simulation of unconstrained spread (no bandwidth constraints). The top graph is a semi-log plot and the bottom graph uses linear scale.

6. EXPERIMENTS

We evaluated the performance of DIBS/TRAFEN by running simulations with varying degrees of router participation, feeding the simulated ICMP-T3 Unreachable traffic into the working DIBS/TRAFEN prototype, and measuring how many simulated machines were infected at the time that DIBS/TRAFEN detected the propagating worm. Figure 8a shows the results for a simulated Code Red v2 worm, while Figure 8b shows the results for a simulated Sapphire/Slammer worm.⁷ For the Code Red worm, we did one set of runs with background scanning noise and one set of runs without noise. Due to time constraints with the Slammer worm, we did only a set of runs without noise, but the Code Red noise results generalize to the Slammer case. For all runs, the simulation parameters were the same as in the validation experiments. Specifically, the address space was the Internet or IPv4 address space of 2^{32} addresses; the simulated Code Red v2 and Sapphire/Slammer worms scanned at rates of 5.65 and 4000 scans per second respectively; there were 384,000 and 120,000 vulnerable machines respectively; the address space utilization, or percentage of reachable addresses, in each of the detector networks was 50%; and the background

⁷We did not perform experiments with the real-world Slammer and Code Red v2 datasets obtained by different organizations at their individual network boundaries, since most of this data does not allow an easy mapping to ICMP-T3 messages. In addition, ICMP-T3 messages from only a single site generally will not provide good detection performance, although we do hope to combine some of the real-world datasets for future experiments.

scanning rate, for the one Code Red experiment, was 1.41 coincidental probes to the Web service port *per Class B network per second* (as observed in the CAS data). We selected a 50% reachability, which is significantly higher than most real Class B networks, to be conservative and ensure that our evaluation did not produce more ICMP-T3 messages than would be observed on the real Internet.

For DIBS, the relevant parameters are N_{DIBS} and Δt_{DIBS} , the number of ICMP-T3 messages per scan alert and the size of the history window respectively. A lower value of N_{DIBS} increases the chances of false positives, since ICMP-T3 background noise can propagate into the scan alerts, and any value below $N_{DIBS} = 4$ makes the system unusable. Conversely, although higher values will lead to more accurate detection, the moment of detection will be later, possibly *too* late. Initial experimentation has shown that $5 \leq N_{DIBS} \leq 15$ gives the best results, and that the value of N_{DIBS} scales with the number of instrumented routers. With more routers, we can increase N_{DIBS} to improve noise tolerance, although the rate of improvement drops off rapidly as N_{DIBS} increases, making values beyond 15 unnecessary.

Similarly, smaller values for Δt_{DIBS} will give a very inaccurate view of events, since alerts on fast scanning IP addresses will be frequently re-issued, and slower scanning worms will not be detected at all. Higher values of Δt_{DIBS} , however, place a serious performance penalty on the analysis system since all packets need to be stored for a longer period of time. Initial experimentation has shown that $\Delta t_{DIBS} > 300$ seconds gives the best results, as values below 300 lead to too many duplicate alerts and hence noise in the scan alert data. Further experiments are needed to examine memory and CPU usage as Δt_{DIBS} and the number of incoming ICMP-T3 messages per second increases, and to determine the best strategy for dividing the analysis across multiple processors.

For the experiments in Figure 8, N_{DIBS} was set to 5, and Δt_{DIBS} was set to 7200 and 3600 for the Code Red and Slammer run respectively. The smaller value for Slammer, taking into account Slammer's faster scan rate, makes the Slammer experiments run faster on our memory-limited hardware, but does not affect the detection results. Given the propagation speed of Slammer and Code Red v2, 3600 and 7200 are significantly larger than intuitively necessary, but the primary purpose of these values is prevent the generation of multiple alerts for the same machine and port combination within too short a time period. These values do not depend on the worm propagation speed, except that they must be significantly larger than the time it takes the worm to find and infect a single vulnerable machine. With values up to 7200, the current system can not detect *slow-scanning* worms that take days or weeks, rather than hours or minutes, to propagate.

DIBS is only half of the system, and the worm detection itself occurs in TRAFEN. In our current prototype, the process model is quite simple, but has given good results. In particular, if two scan alerts (for the same target port) occur within ten seconds of each other, TRAFEN assigns a time likelihood of 1.0 that the two scans are related; if the two scan alerts are more than 300 seconds apart, TRAFEN assigns a time likelihood of 0.0; and if the two scan alerts are between 10.0 and 300.0 seconds apart, TRAFEN assigns a time likelihood scaled linearly between 1.0 and 0.0. The time likelihood is then combined with a port likelihood of 0.9

for the same target port and 0.0 for different target ports, with the port likelihood weighted at three times the time likelihood. The same rules are used for both Slammer and Code Red v2, and are generally applicable worms propagating within hours or days. This range is not appropriate for all noise levels or slower scanning speeds, however, an issue that must be addressed with future work. Overall, the three rules capture the fact that a propagating worm generates more and more scan alerts as it infects more machines, while also capturing the fact that two scans on the same port are not necessarily related.

With the simulation and DIBS parameters and TRAFEN ruleset above, we obtain the results in Figure 8. In the case of Code Red v2, when there were instrumented routers covering two Class B networks, DIBS/TRAFEN detected the worm before it had infected 0.2% of the vulnerable machines. The infection percentage rapidly dropped to 0.03% for a coverage of sixteen Class B networks, and then remained roughly steady as the number of Class B networks increased further. Similarly, for Sapphire/Slammer, DIBS/TRAFEN detected the worm at an infection percentage of 0.01% for four Class B networks, and an infection percentage of 0.005% for sixteen Class B networks. An important difference is that DIBS/TRAFEN did not detect the Sapphire/Slammer worm at all when the coverage was only two Class B networks. Since Sapphire/Slammer propagates so quickly, and routers are configured to send only a few ICMP-T3 messages per second, DIBS/TRAFEN simply does not receive enough ICMP-T3 messages when there are only two instrumented routers.

For Code Red v2, we see that the detection results are the same when background noise is present. In addition, there were *no* false positives during a simulated 16-hour period before the worm was launched. Although the signal-to-noise ratio in the collected ICMP-T3 messages is high, DIBS must see several probes from the same source address before generating an alert. This happens only rarely with the noise level observed at TRIUMF Canada, and thus the signal-to-noise ratio in the DIBS *alerts* is good enough to prevent false positives. Background noise remains a critical concern, however, since we can expect that DIBS/TRAFEN will encounter more background noise in production use than what was observed at TRIUMF Canada. In addition, the background noise in the simulation was distributed uniformly across the space of source IP addresses, the opposite of what would be observed if a small group of attackers was conducting scans from a small set of source machines.

Fortunately, more complex models can address the noise problem, allow detection of slower-scanning worms, and allow faster detection of worms that bias their random scanning toward local addresses. We currently are evaluating the detection capabilities and scalability of approaches in which linear or exponential growth curves are matched against the scan alerts, as well as of Hidden Markov Models [19]. The advantage of the simulation framework is that each new approach can be evaluated against an Internet-sized address space without doing a full, real-world deployment, and the advantage of the TRAFEN framework is that each new approach is simply a drop-in replacement for the current ruleset, allowing extremely rapid development.

Overall, the results demonstrate the significant promise of worm-detection systems. The system can detect worms early in their lifetime; the system scales well, since the ICMP-

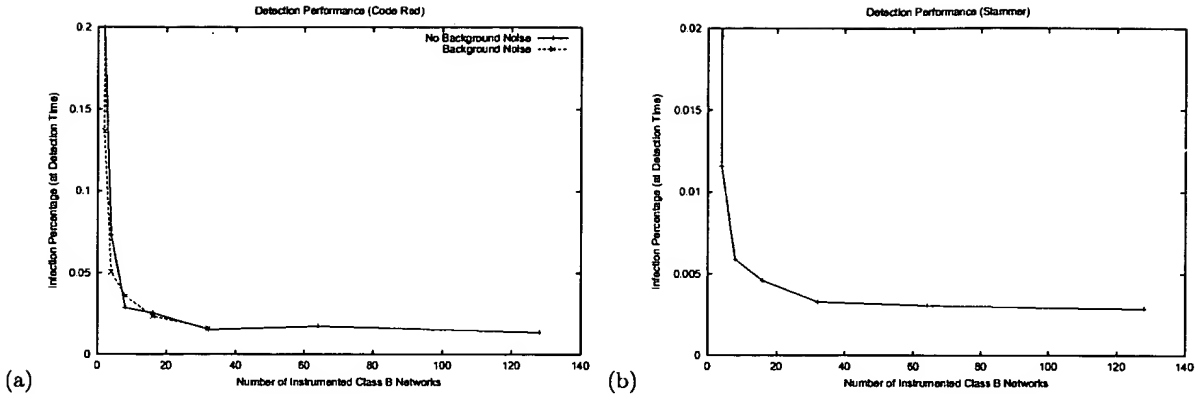


Figure 8: The percentage of vulnerable machines infected at detection time for (a) a simulated Code Red v2 worm and (b) a simulated Sapphire/Slammer worm. Each data point is an average of three simulation runs. Two Class B networks corresponds to an address-space coverage of $2^{17}/2^{32}$, while 128 Class B networks corresponds to an address-space coverage of $2^{23}/2^{32}$.

T3 messages are a manageable data stream be federated, each copy handling different address or port ranges; and, perhaps most importantly, researchers are beginning to work on automated response mechanisms [17] that could slow or stop even Slammer-like worms after detection.

7. RELATED WORK

Worm Modeling. Kephart and White [13] attempted to model viruses spread by floppy disks using epidemic models in the early nineties, but found that the continuous time deterministic *simple epidemic model* [10] predicted much faster virus spread than observed. They hypothesized that this was because people only exchange disks with other people they know, while the model assumes equal probability of interaction between all individuals (the *homogeneous mixing assumption*). Staniford [21] appears to have been the first to propose modeling the Code Red worm using a model that was essentially the same as the “simple epidemic model”, and, in this case, it works well since random uniform scanning corresponds well with the homogeneous mixing assumption. It has since been used in several studies [22, 16, 14, 26]. However, the literature on epidemic models (e.g., [10]) includes many other variants, including stochastic formulations, discrete time models, models that include removals from the infected population—in the context of worms, this would correspond to patching, reboots, or scan filtering—and models of vector-borne diseases. The *general epidemic model*, which adds removals, has been considered in [14, 26, 17] using the continuous time formulation, while [6] considers a discrete time model.

Our work has most in common with [17, 6, 27], where models are used to study the feasibility of detection/defense system designs, but we go further since we test a working prototype system. The macroscopic level of our mixed abstraction level simulation model appears to share some common traits with the model used in [17], in terms of epidemics and topology considered, but leveraging off the SSFNet simulator we also provide packet-level capabilities.

Worm Detection Systems. There has been some recent work on the detection of Internet worms. The distributed

NetBait system [7] does not provide automated detection of previously unknown worms, since it relies on the availability of signatures for extracting probe data from available log-files. After the development of a signature through some other means, however, administrators can use NetBait to identify infected machines, and analyze the nature and extent of the epidemic.

Zou et al. have developed a worm-detection approach based on Kalman filters [27]. The system collects scan alerts from distributed ingress and egress monitors, and applies a Kalman filter to the scan alerts to see if the pattern of scanning activity matches their SIR-based model of worm propagation. For an address space with 2^{32} addresses (i.e., the Internet), monitoring coverage of $2^{17}/2^{32}$, and 500,000 vulnerable machines, their system can detect a simulated Code Red worm, and predict its overall infection rate, as soon as the worm infects approximately 5% of the vulnerable machines. Our ability to detect a Code Red infection at a lower percentage arises from our use of ICMP-T3 Unreachable messages from multiple instrumented routers, which allows us to detect even the scanning activity that hits any particular network only once, rather than waiting until the scanning activity has hit a single network enough times to be considered significant. The ingress and egress filters, or other distributed intrusion-detection systems, could provide DIBS/TRAFEN with significant additional data, however. In addition, the Kalman filter, which has several attractive features, could improve the performance of our current rule-set.

Moore, Voelker and Savage [18] use the same underlying router behavior on which we rely, and collect ICMP-T3 messages and other data to detect the “backscatter” from denial-of-service attacks. Although their system is not directly applicable to worm detection, it does illustrate that ICMP-T3 messages can be collected and used for many purposes, amortizing the cost of collecting those messages in the first place. Yagneswaran, Barford and Ullrich analyze characteristics of worm and non-worm traffic, many of which we already use, and others that might influence the development of future detection models [25]. Yagneswaran also considers passive monitoring of unused blocks of address space,

while our system can monitor both used and unused address blocks with appropriate router placements.

8. CONCLUSIONS AND FUTURE WORK

We have described a simulation model that combines coarse and fine grained elements to model detailed effects of large-scale worm attacks, and we have shown its usefulness in generating test traffic for the DIBS/TRAFEN detection system prototype. We used a spatially distributed stochastic version of the continuous time general epidemic model and found it useful for modeling the early stages of the Code Red v2 and Sapphire/Slammer worms' spread.

Our experiments with the Code Red and Slammer models indicate that even the relatively simple algorithms currently implemented in the DIBS/TRAFEN system could have *i)* detected the Code Red worm before it had infected 0.2% of the vulnerable hosts while monitoring only two Class B networks, and *ii)* detected the Slammer worm at 0.01% infection while monitoring at least four class B networks. (In both cases detection was further improved with increased coverage.) These results are quite encouraging and work is continuing on the algorithms to reduce the risk of false positives in the face of background scanning noise (although this did not present a problem in a test that included a background noise level observed just before the launch of the Code Red v2 worm).

Future work includes taking advantage of the distributed execution capabilities of SSFNet to add details and scale to the worm model, and to conduct more experiments with hypothetical worm scenarios.

ACKNOWLEDGMENTS

We thank Ken Eichman at the Chemical Abstract Service and Andrew Daviel at TRIUMF Canada for generously providing the Code Red and Sapphire/Slammer data sets, respectively.

This research is supported in part by DARPA Contracts N66001-96-C-8530, F30602-00-2-0585, NSF Grants ANI-98 08964, EIA-98-02068, CISE-0209144, and Dept. of Justice contract 2000-CX-K001. Points of view in this document are those of the authors and do not necessarily represent the official position of the United States Department of Justice.

9. REFERENCES

- [1] Labrea. <http://www.hackbusters.net/LaBrea>.
- [2] Ssfnet web site. <http://www.ssfnet.org/>.
- [3] F. Baker. Rfc 1812: Requirements for IP version 4 routers. Request for Comments 1812, June 1995.
- [4] Vincent Berk, Wayne Chung, Valentino Crespi, George Cybenko, Robert Gray, Diego Hernandez, Guofei Jiang, Han Li, and Yong Sheng. Process Query Systems for Surveillance and Awareness. In *Proceedings of the SCI 2003*, Orlando, Florida, July 2003.
- [5] Vincent H. Berk, Robert S. Gray, and George Bakos. Using Sensor Networks and Data Fusion for Early Detection of Active Worms. In *Proceedings of AeroSense 2003: SPIE's 17th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls*, Orlando, Florida, April 2003.
- [6] Z. Chen, L. Gao, and K. Kwiat. Modeling the Spread of Active Worms. INFOCOM 2003, 2003.
- [7] Brent N. Chun, Jason Lee, and Hakim Weatherspoon. Netbait: A distributed worm detection service. Available at <http://netbait.plain-lab.org/>, 2003.
- [8] Cisco. Dealing with mallocfail and high CPU utilization resulting from the "Code Red" worm. <http://www.cisco.com/warp/public/-63/ts.codred.worm.shtml>, October 2001.
- [9] J. Cowie, D. Nicol, and A. Ogielski. Modeling the Global Internet. *IEEE Computing in Science and Engineering*, 1(1):42-50, Jan.-Feb. 1999.
- [10] D.J. Daley and J. Gani. *Epidemic Modelling: An Introduction*. Cambridge University Press, Cambridge, UK, 1999.
- [11] Silicon Defense. CounterMalice—Worm Containment System. <http://www.silicondefense.com/products/countermalice/>, 2003.
- [12] S. Floyd and V. Paxson. Difficulties in Simulating the Internet. *IEEE/ACM Transactions on Networking*, 9(4):392-403, August 2001.
- [13] J.O. Kephart and S.R. White. Measuring and Modeling Computer Virus Prevalence. Proceedings of the 1993 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, CA, May 1993.
- [14] M. Liljenstam, Y. Yuan, B.J. Premore, and D. Nicol. A Mixed Abstraction Level Model of Large-Scale Internet Worm Infestations. in Proc. of the Tenth IEEE/ACM Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), Fort Worth, TX, Oct 2002. IEEE Computer Society Press.
- [15] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. Inside the Slammer Worm. *IEEE Security and Privacy*, 1(4):33-39, July 2003.
- [16] D. Moore, C. Shannon, and K. Claffy. Code-Red: A case study on the spread and victims of an Internet worm. in Proc. of the Internet Measurement Workshop (IMW), Marseille, France, Nov 2002. ACM Press.
- [17] David Moore, Colleen Shannon, Geoffrey M. Voelker, and Stefan Savage. Internet quarantine: Requirements for containing self-propagating code. In *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003)*, April 2003.
- [18] David Moore, Geoffrey M. Voelker, and Stefan Savage. Inferring Internet Denial-of-Service activity. In *Proceedings of the 10th USENIX Security Symposium (USENIX'01)*, Washington, DC, August 2001.
- [19] Lawrence R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceeding of the IEEE*, 77, Num. 2:257-286, 1989.
- [20] Donald B. Reid. An algorithm for Tracking Multiple Targets. *IEEE Transactions on Automatic Control*, AC-24, Num. 6:843-854, 1979.
- [21] S. Staniford. Code Red Analysis Pages: July infestation analysis. <http://www.silicondefense.com/cr/july.html>, 2001.
- [22] S. Staniford, V. Paxson, and N. Weaver. How to Own the Internet in Your Spare Time. in Proc. of the USENIX Security Symposium, 2002. <http://www.icir.org/vern/papers/cdc-usenix-sec02/index.html>.
- [23] A. Turner and M. Bing. tcpreplay project page (sourceforge). <http://tcpreplay.sourceforge.net/>, 2003.
- [24] I. van Beijnum. *BGP*. O'Reilly & Associates, Sebastopol, CA, 2002.
- [25] Vinod Yagneswaran, Paul Barford, and Johannes Ullrich. Internet intrusions: Global characteristics and prevalence. In *Proceedings of the International Conference on Measurements and Modeling of Computer Systems (SIGMETRICS 2003)*, San Diego, California, June 2003.
- [26] C. Zou, L. Gao, W. Gong, and D. Towsley. Code Red Worm Propagation Modeling and Analysis. 9th ACM Conference on Computer and Communication Security (CCS), Washington DC, Nov 2002.
- [27] Cliff C. Zou, Lixin Gao, Weibo Gong, and Don Towsley. Monitoring and early warning for internet worms. Technical Report TR-CSE-03-01, University of Massachusetts at Amherst, 2003.

Resource Center:

[Linux](#) | [Home/Home Office](#) | [Convergence](#) | [Enterprise](#) | [IT Outsourcing](#)

**Enterprise Customer Expectation
& Satisfaction Survey 2003**
- A Voice@Data Report

Download PDF
for only US\$22/
Rs.1000/-
[CLICK HERE](#)

Search in [Entire CIOLOGO Network](#) [Advanced Search](#)

[Home](#) | [Site Map](#) | [Shopping](#) | [Travel](#) | [Training](#) | [Help](#) | [Find a Job](#) | [Get Free IT Info](#) | [Recommend this site](#)

[Home](#) > [Technology and You](#) > [Techie touch](#) > Virus, worm, what's the difference?



Today's News

- [Cyber cafe for the visually challenged](#)
- [Satyam, Hummingbird tie up for Singapore facility](#)
- [ThirdDream: Nihilent's offering for SMEs](#)
- [Amazon to feature Google search and ad links](#)
- [Ex-Intel employee appeals court to defend spam](#)

[More news...](#)

Site Guide

- [News & Features](#)
- [Money & Markets](#)
- [Opinions](#)
- [Flavour of the Month](#)
- [eEnterprise](#)
- [Building Blocks](#)
- [Enterprise Processes](#)
- [Cutting Edge](#)
- [Tutorials](#)
- [Developer Plus](#)
- [Technology & You](#)
- [PC Help](#)
- [Digital Life](#)

Virus, worm, what's the difference?

After a bout of the Love Bug, no computer user can really afford to be ignorant about virus attacks and their painful aftermath. So it is time we traced the origin and variants of these cryptic destroyers.

[Krithi Aiyappa](#)

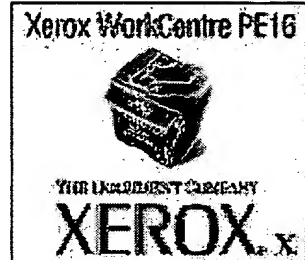
Tuesday, July 25, 2000

With the advent of computers and the Internet, the term 'vital information resources under siege', better known as VIRUS, has taken on an entirely different connotation. It has become regular fare to read about virus attacks and the consequent losses running into billions of dollars. Now it is not just the term virus that brings the shivers to computer users in general and software professionals in particular. There are others too, for instance Trojan Horse and Internet worms that trigger a similar reaction. One might think that a virus is a virus whatever you may call it. But there is a world of difference between a virus and a worm—these are as different as the biological virus and the bacteria—because the effects are different and so are the remedies. In fact, the word virus in Latin means poison.

So, what is a computer virus? It is a program or a block of executable code written to surreptitiously enter your computer and infect your files by attaching it to, overwriting or replacing another program. A computer virus is not self generated but must be written by someone with a specific purpose in mind. Typically, a virus has two functions:

- Self-replication and propagation – to spread itself from one file to another by creating either the exact or modified copies of itself, wherein the replication is intentional and not a side effect
- Delivering payload – that is implementing the damage planned by the perpetrator.

Advertisement



CIOLOGO SPECIALS

TRENDS

- [Sun Tech Days 2003](#)

A computer virus behaves in the same way as its biological counterpart. When a program infected by a virus is running, the virus code gets a chance to inspect

- [NASSCOM 2003](#)
- [The Tablet PC Special](#)
- [The antitrust case](#)
- [Bangalore IT.COM](#)
- [JAS Quarterly Results](#)
- [Internet Special](#)
- [ITES](#)
- [VoIP](#)
- [Internet Telephony](#)
- ENTERPRISE**
- [Data Warehousing](#)
- [Knowledge Management](#)
- [Servers](#)
- [eProcurement](#)
- [Storage](#)
- [Outsourcing](#)
- [Case Studies](#)



CIOL Services

[Ask Tech Expert](#) | [Training](#)
[Events](#) | [IT Jobs](#) | [Travel](#) |
[IT Outsourcing](#) | [the DQweek Dealer Online](#)
[Community](#) | [IT Shopping](#) |
[Computers@Home Product Mart](#)

its environment and look for loopholes in the host and infect other program files, which in turn become carriers. A virus could either be benign or malignant. A benign virus does no real damage to your system; it may do nothing more than display a message on a particular date or time. Nevertheless, it hogs disk space and memory by using up CPU processing time, and money is wasted in its detection and removal. A malignant virus on the other hand is a program written with malicious intent to wreak havoc on your computer, at times doing more damage than the perpetrator had originally planned. It could alter programs, write incorrect information into files, delete files or even erase your entire hard disk.

So when, where and by whom was this Pandora's box opened? Initially, these self replicating programs were really not viruses but were known as rabbits, which took over the computing time of the machine leaving no time or space for other programs. These were first written in the 1960s. Then in a case of life taking after fiction, came the worms in the 1970s, inspired by a book called *The Shockwave*, wherein the writer talks about a program that replicates segments of itself all over a network. The first worm that was created was called the creeper and the anti virus, what else, the reaper. This opened the Pandora's box, quite literally. The creeper would run on one system, copy itself onto another and delete itself on the original. This was later modified to enable the program's replication and migration. The reaper on the other hand would move through the system destroying all copies of the creeper and then deletes itself. The first true virus that was created was a part of a research project for Apple II computer in the 1980s called 'Elk Cloner'.

Another book called *Neuromancer* glorifying computer hackers was the driving force behind a cult that came to be known as Cyberpunk. Robert T Morris, one such person influenced by the book, went on to write the infamous 'Internet worm'.

A Trojan Horse is an elementary form of a malicious code written to perform some covert act. It appears to perform some useful or entertaining function like displaying a screen saver or a message while concealing a destructive purpose, hence the name, taken after the Greek legend. It differs from a virus in that it does not replicate itself and does not propagate. A dropper is a program that installs a Trojan or virus covertly.

A worm on the other hand resides in active memory, replicates itself and duplicates itself over computer networks. The various features used by an operating

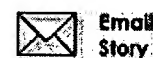


Search



[Home](#)

Sorry, there is no ciol.c incorrectly, or that the p



system to make its functions automatic and invisible to the user are in turn used by the worm to its advantage. Unlike a virus it does not attach itself to a host program.



Virus attacks are a very real threat faced by the computer industry and requires stringent laws to curb its widespread effects. These attacks could tamper with sensitive information and paralyze entire organizations. In the next part we'll take a look at what precautions can be taken to prevent these attacks and what to do when you are under attack.



Message boards

Discuss this and many other IT topics at the [CIOL message board](#)

Previous Stories

[Lights... camera... action](#)

[I've got the power!](#)

[Say cheese to digital photography](#)



Page(s) 1

Today's News

- [Cyber cafe for the visually challenged](#)
- [Satyam, Hummingbird tie up for Singapore facility](#)
- [ThirdDream: Nihilent's offering for SMEs](#)
- [Amazon to feature Google search and ad links](#)
- [Ex-Intel employee appeals court to defend spam](#)

[More news...](#)



Other Cyber Media web sites

- | | | | |
|---------------------------------------|---------------------------------------|-------------------------------|--------------------------------------|
| [Dataquest] | [Voice&Data] | [PCQuest] | [Computers@Home] |
| [DQ Channels India] | [IDC India] | [Training] | [CIOL Shop] |
| [the DQweek] | [CIOL Jobs] | [Cyberexpo] | [Cyber Multimedia] |
| [Cyber Media India] | [GlobalOutsourcing] | [Travel] | [Cyber Astro] |
| [BioSpectrum] | | | |

About CIOL

[How to advertise](#) | [Custom Publishing](#) | [Contact us](#) | [Privacy policy](#) | [Jobs@CIOL](#) | [Write for CIOL](#)



Copyright © [Cyber India Online Limited](#). All rights reserved.

Reproduction in whole or in part in any form or medium without express written permission is prohibited.

Usage of this web site is subject to [terms and conditions](#).

Broken links? Problems with site? Send email to webmaster@ciol.com



US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

Search: ☒ The ACM Digital Library ☐ The Guide

(fsm and monitor* and simulat* and network* and securit* an

[SEARCH](#)

THE ACM DIGITAL LIBRARY

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used

fsm and **monitor** and **simulat** and **network** and **securit** and **fault** and **histor** and **predict**

Found **18,983** of
124,098

Sort results
by

[Save results to a Binder](#)

Try an [Advanced Search](#)

Display
results

[Search Tips](#)

Try this search in [The ACM Guide](#)

☐ Open results in a new
window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐

1 [Real-time protocol analysis for detecting link-state routing protocol attacks](#)

Ho-Yen Chang, S. Felix Wu, Y. Frank Jou

February 2001 **ACM Transactions on Information and System Security (TISSEC)**, Volume 4
Issue 1

Full text available: [pdf\(252.10 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

A real-time knowledge-based network intrusion-detection model for a link-state routing protocol is presented for the OSPF protocol. This model includes three layers: a data process layer to parse packets and dispatch data; and event abstractor to abstract predefined real-time events for the link-state routing protocol; and an extended timed finite state machine to express the real-time behavior of the protocol engine and to ...

Keywords: OSPF attacks, event correlation, knowledge-based IDS, link-state routing protocol security, real-time misuse intrusion detection, real-time network protocol analysis, timed finite state machine

2 [Evolutionary design of complex software \(EDCS\) demonstration days 1999](#)

Wayne Stidolph

January 2000 **ACM SIGSOFT Software Engineering Notes**, Volume 25 Issue 1

Full text available: [pdf\(1.90 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [index terms](#)

This report summarizes the Product/Technology demonstrations given at Defense Advanced Research Projects Agency (DARPA) Evolutionary Design of Complex Software (EDCS) Program Demonstration Days, held 28-29 June 1999 at the Sheraton National Hotel, Arlington, VA.

3 [Computing curricula 2001](#)

September 2001 **Journal on Educational Resources in Computing (JERIC)**

Full text available: [pdf\(613.63 KB\)](#)

[html\(2.78 KB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

4 [Fast detection of communication patterns in distributed executions](#)

Thomas Kunz, Michiel F. H. Seuren

November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research**

Full text available:  pdf(4.21 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

5 Astrolabe: A robust and scalable technology for distributed system monitoring, management, and data mining 

Robbert Van Renesse, Kenneth P. Birman, Werner Vogels

May 2003 **ACM Transactions on Computer Systems (TOCS)**, Volume 21 Issue 2

Full text available:  pdf(341.62 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Scalable management and self-organizational capabilities are emerging as central requirements for a generation of large-scale, highly dynamic, distributed applications. We have developed an entirely new distributed information management system called Astrolabe. Astrolabe collects large-scale system state, permitting rapid updates and providing on-the-fly attribute aggregation. This latter capability permits an application to locate a resource, and also offers a scalable way to track sys ...

Keywords: Aggregation, epidemic protocols, failure detection, gossip, membership, publish-subscribe, scalability

6 Data integrity: Web application security assessment by fault injection and behavior monitoring 

Yao-Wen Huang, Shih-Kun Huang, Tsung-Po Lin, Chung-Hung Tsai

May 2003 **Proceedings of the twelfth international conference on World Wide Web**

Full text available:  pdf(4.53 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

As a large and complex application platform, the World Wide Web is capable of delivering a broad range of sophisticated applications. However, many Web applications go through rapid development phases with extremely short turnaround time, making it difficult to eliminate vulnerabilities. Here we analyze the design of Web application security assessment mechanisms in order to identify poor coding practices that render Web applications vulnerable to attacks such as SQL injection and cross-site scr ...

Keywords: black-box testing, complete crawling, fault injection, security assessment, web application testing

7 Network interactions: Simulating realistic network worm traffic for worm warning system design and testing 

Michael Liljenstam, David M. Nicol, Vincent H. Berk, Robert S. Gray

October 2003 **Proceedings of the 2003 ACM workshop on Rapid Malcode**

Full text available:  pdf(308.41 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Reproducing the effects of large-scale worm attacks in a laboratory setup in a realistic and reproducible manner is an important issue for the development of worm detection and defense systems. In this paper, we describe a worm simulation model we are developing to accurately model the large-scale spread dynamics of a worm and many aspects of its detailed effects on the network. We can model slow or fast worms with realistic scan rates


on realistic IP address spaces and selectively model local d ...

Keywords: code red, network modeling and simulation, network security, slammer, worm detection systems, worms

8 Special feature: Report on a working session on security in wireless ad hoc networks

Levente Buttyán, Jean-Pierre Hubaux

January 2003 **ACM SIGMOBILE Mobile Computing and Communications Review**, Volume 7 Issue 1

Full text available:  pdf(2.50 MB)

Additional Information: [full citation](#), [references](#)



9 Verisim: Formal analysis of network simulations

Karthikeyan Bhargavan, Carl A. Gunter, Moonjoo Kim, Insup Lee, Davor Obradovic, Oleg Sokolsky, Mahesh Viswanathan

August 2000 **ACM SIGSOFT Software Engineering Notes , Proceedings of the International Symposium on Software Testing and Analysis**, Volume 25 Issue 5

Full text available:  pdf(325.27 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)




Why are there so few successful "real-world" programming and testing tools based on academic research? This talk focuses on program analysis tools, and proposes a surprisingly simple explanation with interesting ramifications. For a tool aimed at developers or testers to be successful, people must use it - and must use it to help accomplish their existing tasks, rather than as an end in itself. If the tool does not help them get their job done, or the effort to learn and/or use th ...

10 Illustrative risks to the public in the use of computer systems and related technology

Peter G. Neumann

January 1996 **ACM SIGSOFT Software Engineering Notes**, Volume 21 Issue 1

Full text available:  pdf(2.54 MB)

Additional Information: [full citation](#)



11 Survey of recent operating systems research, designs and implementations

C. Mohan

January 1978 **ACM SIGOPS Operating Systems Review**, Volume 12 Issue 1

Full text available:  pdf(2.54 MB)

Additional Information: [full citation](#), [references](#)



12 Cluster resource management: An integrated experimental environment for distributed systems and networks

Brian White, Jay Lepreau, Leigh Stoller, Robert Ricci, Shashi Guruprasad, Mac Newbold, Mike Hibler, Chad Barb, Abhijeet Joglekar

December 2002 **ACM SIGOPS Operating Systems Review**, Volume 36 Issue SI

Full text available:  pdf(2.10 MB)


Additional Information: [full citation](#), [abstract](#), [references](#)



Three experimental environments traditionally support network and distributed systems research: network emulators, network simulators, and live networks. The continued use of multiple approaches highlights both the value and inadequacy of each. Netbed, a descendant of Emulab, provides an experimentation facility that integrates these approaches, allowing researchers to configure and access networks composed of emulated, simulated, and wide-area nodes and links. Netbed's primary goals are ease ...

13 Process migration

September 2000 **ACM Computing Surveys (CSUR)**, Volume 32 Issue 3

Full text available:  pdf(1.24 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Process migration is the act of transferring a process between two machines. It enables dynamic load distribution, fault resilience, eased system administration, and data access locality. Despite these goals and ongoing research efforts, migration has not achieved widespread use. With the increasing deployment of distributed systems in general, and distributed operating systems in particular, process migration is again receiving more attention in both research and product development. As hi ...

Keywords: distributed operating systems, distributed systems, load distribution, process migration



14 Network management using expert diagnostics

Wayne Fuller

August 1999 **International Journal of Network Management**, Volume 9 Issue 4

Full text available:  pdf(1.45 MB)

Additional Information: [full citation](#), [abstract](#), [index terms](#)

Networks have become a key component of the corporate infrastructure. Managing the networks, which often carry a diverse set of information & lpar; e.g. voice, data, video) over a diverse set of media & lpar; e.g. wire, cable, RF) with a mixture of owned and leased transmission assets that are often geographically distributed and run a diverse set of protocols, is a major challenge. One of the most promising techniques applies expert system approaches to the management of networks. Co ...



15 Knowledge based fault management for OSI networks

Celia A. Joseph, A. Sherzer, K. Muralidhar

June 1990 **Proceedings of the third international conference on Industrial and engineering applications of artificial intelligence and expert systems - Volume 1**

Full text available:  pdf(826.21 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


The OSI Fault Management system (OSIFaM) is an evolving knowledge-based system for fault management of Open System Interconnection (OSI) networks. Our goal is to develop a knowledge-based tool that will reduce the expertise needed to recognize, diagnose and correct faults in OSI networks. For our first implementation, we are focusing on MAP 3.0 networks. This paper provides an overview of fault management in general, a brief survey of other fault management developments, the characteristics ...



16 IS '97: model curriculum and guidelines for undergraduate degree programs in information systems

Gordon B. Davis, John T. Gorgone, J. Daniel Couger, David L. Feinstein, Herbert E. Longenecker

December 1997 **ACM SIGMIS Database , Guidelines for undergraduate degree programs on Model curriculum and guidelines for undergraduate degree programs in information systems**, Volume 28 Issue 1

Full text available:  pdf(7.24 MB)

Additional Information: [full citation](#), [citations](#)



17 Session summaries from the 17th symposium on operating systems principle (SOSP'99)



Jay Lepreau, Eric Eide

April 2000 **ACM SIGOPS Operating Systems Review**, Volume 34 Issue 2

Full text available:  [pdf\(3.15 MB\)](#) Additional Information: [full citation](#), [index terms](#)

18 Summary of the sigmetrics symposium on parallel and distributed processing 

Jeffrey K. Hillingsworth, Barton P. Miller

March 1999 **ACM SIGMETRICS Performance Evaluation Review**, Volume 26 Issue 4

Full text available:  [pdf\(1.17 MB\)](#) Additional Information: [full citation](#), [index terms](#)

19 Fault detection with multiple observers 

Clark Wang, Mischa Schwartz

February 1993 **IEEE/ACM Transactions on Networking (TON)**, Volume 1 Issue 1

Full text available:  [pdf\(854.36 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

20 File and storage systems: Preserving peer replicas by rate-limited sampled voting 

Petros Maniatis, David S. H. Rosenthal, Mema Roussopoulos, Mary Baker, TJ Giuli, Yanto Muliadi

October 2003 **Proceedings of the nineteenth ACM symposium on Operating systems principles**

Full text available:  [pdf\(336.27 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The LOCKSS project has developed and deployed in a world-wide test a peer-to-peer system for preserving access to journals and other archival information published on the Web. It consists of a large number of independent, low-cost, persistent web caches that cooperate to detect and repair damage to their content by voting in "opinion polls." Based on this experience, we present a design for and simulations of a novel protocol for voting in systems of this kind. It incorporates rate limitation an ...





Keywords: digital preservation, rate limiting, replicated storage

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2003 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)



US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

Search: ☒ The ACM Digital Library ☐ The Guide

(fsm and monitor* and simulat* and network* and securit* an

SEARCH

THE ACM DIGITAL LIBRARY

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used

fsm and **simulat** and **network** and **securit** and **fault** and **histor** and **predict**

Found 18,595 of 124,098

Sort results by

relevance

[Save results to a Binder](#)

[Try an Advanced Search](#)

Display results

expanded form

[Search Tips](#)

[Try this search in The ACM Guide](#)

☐ Open results in a new window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐

1 [Real-time protocol analysis for detecting link-state routing protocol attacks](#)

Ho-Yen Chang, S. Felix Wu, Y. Frank Jou

February 2001 **ACM Transactions on Information and System Security (TISSEC)**, Volume 4 Issue 1

Full text available: pdf(252.10 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

A real-time knowledge-based network intrusion-detection model for a link-state routing protocol is presented for the OSPF protocol. This model includes three layers: a data process layer to parse packets and dispatch data; and event abstractor to abstract predefined real-time events for the link-state routing protocol; and an extended timed finite state machine to express the real-time behavior of the protocol engine and to ...

Keywords: OSPF attacks, event correlation, knowledge-based IDS, link-state routing protocol security, real-time misuse intrusion detection, real-time network protocol analysis, timed finite state machine

2 [A composable framework for secure multi-modal access to internet services from Post-PC devices](#)

Steven J. Ross, Jason L. Hill, Michael Y. Chen, Anthony D. Joseph, David E. Culler, Eric A. Brewer

October 2002 **Mobile Networks and Applications**, Volume 7 Issue 5

Full text available: pdf(340.33 KB)



Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The Post-PC revolution is bringing information access to a wide range of devices beyond the desktop, such as public kiosks, and mobile devices like cellular telephones, PDAs, and voice based vehicle telematics. However, existing deployed Internet services are geared toward the secure rich interface of private desktop computers. We propose the use of an infrastructure-based secure proxy architecture to bridge the gap between the capabilities of Post-PC devices and the requirements of Internet ser ...

Keywords: internet, middleware, post-PC, security, transcoding

3 [Computing curricula 2001](#)

September 2001 **Journal on Educational Resources in Computing (JERIC)**

Full text available:  [pdf\(613.63 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)
 [html\(2.78 KB\)](#)

4 Evolutionary design of complex software (EDCS) demonstration days 1999 

Wayne Stidolph

January 2000 **ACM SIGSOFT Software Engineering Notes**, Volume 25 Issue 1


Full text available:  [pdf\(1.90 MB\)](#) Additional Information: [full citation](#), [abstract](#), [index terms](#)

This report summarizes the Product/Technology demonstrations given at Defense Advanced Research Projects Agency (DARPA) Evolutionary Design of Complex Software (EDCS) Program Demonstration Days, held 28-29 June 1999 at the Sheraton National Hotel, Arlington, VA.

5 On visual formalisms 

David Harel


May 1988 **Communications of the ACM**, Volume 31 Issue 5

Full text available:  [pdf\(1.59 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

The higraph, a general kind of diagramming object, forms a visual formalism of topological nature. Higraphs are suited for a wide array of applications to databases, knowledge representation, and, most notably, the behavioral specification of complex concurrent systems using the higraph-based language of statecharts.

6 Process migration 

September 2000 **ACM Computing Surveys (CSUR)**, Volume 32 Issue 3

Full text available:  [pdf\(1.24 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Process migration is the act of transferring a process between two machines. It enables dynamic load distribution, fault resilience, eased system administration, and data access locality. Despite these goals and ongoing research efforts, migration has not achieved widespread use. With the increasing deployment of distributed systems in general, and distributed operating systems in particular, process migration is again receiving more attention in both research and product development. As hi ...

Keywords: distributed operating systems, distributed systems, load distribution, process migration

7 Fast detection of communication patterns in distributed executions 

Thomas Kunz, Michiel F. H. Seuren

November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research**

Full text available:  [pdf\(4.21 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

8 

Cryptographic protocols/ network security: A composable cryptographic library with

nested operations

Michael Backes, Birgit Pfitzmann, Michael Waidner

October 2003 **Proceedings of the 10th ACM conference on Computer and communication security**

Full text available:  [pdf\(234.97 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

We present the first idealized cryptographic library that can be used like the Dolev-Yao model for automated proofs of cryptographic protocols that use nested cryptographic operations, while coming with a cryptographic implementation that is provably secure under active attacks.

Keywords: cryptographically composable operators, cryptography, security analysis of protocols, simulatability

9 Network interactions: Simulating realistic network worm traffic for worm warning system design and testing

Michael Liljenstam, David M. Nicol, Vincent H. Berk, Robert S. Gray

October 2003 **Proceedings of the 2003 ACM workshop on Rapid Malcode**

Full text available:  [pdf\(308.41 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Reproducing the effects of large-scale worm attacks in a laboratory setup in a realistic and reproducible manner is an important issue for the development of worm detection and defense systems. In this paper, we describe a worm simulation model we are developing to accurately model the large-scale spread dynamics of a worm and many aspects of its detailed effects on the network. We can model slow or fast worms with realistic scan rates on realistic IP address spaces and selectively model local d ...

Keywords: code red, network modeling and simulation, network security, slammer, worm detection systems, worms

10 Special feature: Report on a working session on security in wireless ad hoc networks

Levente Buttyán, Jean-Pierre Hubaux

January 2003 **ACM SIGMOBILE Mobile Computing and Communications Review**, Volume 7 Issue 1

Full text available:  [pdf\(2.50 MB\)](#) Additional Information: [full citation](#), [references](#)

11 Cluster resource management: An integrated experimental environment for distributed systems and networks

Brian White, Jay Lepreau, Leigh Stoller, Robert Ricci, Shashi Guruprasad, Mac Newbold, Mike Hibler, Chad Barb, Abhijeet Joglekar

December 2002 **ACM SIGOPS Operating Systems Review**, Volume 36 Issue SI


Full text available:  [pdf\(2.10 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

Three experimental environments traditionally support network and distributed systems research: network emulators, network simulators, and live networks. The continued use of multiple approaches highlights both the value and inadequacy of each. Netbed, a descendant of Emulab, provides an experimentation facility that integrates these approaches, allowing researchers to configure and access networks composed of emulated, simulated, and wide-area nodes and links. Netbed's primary goals are ease ...

12 Perfectly secure message transmission

Danny Dolev, Cynthia Dwork, Orli Waarts, Moti Yung

January 1993 **Journal of the ACM (JACM)**, Volume 40 Issue 1

Full text available:  pdf(2.42 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)


This paper studies the problem of perfectly secure communication in general network in which processors and communication lines may be faulty. Lower bounds are obtained on the connectivity required for successful secure communication. Efficient algorithms are obtained that operate with this connectivity and rely on no complexity-theoretic assumptions. These are the first algorithms for secure communication in a general network to simultaneously achieve the three goals of perfect secrecy, perfect confidentiality, and perfect integrity.

Keywords: distributed computing, fault-tolerance, perfectly secure communication

13 Illustrative risks to the public in the use of computer systems and related technology

Peter G. Neumann

January 1996 **ACM SIGSOFT Software Engineering Notes**, Volume 21 Issue 1


Full text available:  pdf(2.54 MB)

Additional Information: [full citation](#)

14 Network management using expert diagnostics

Wayne Fuller

August 1999 **International Journal of Network Management**, Volume 9 Issue 4

Full text available:  pdf(1.45 MB)

Additional Information: [full citation](#), [abstract](#), [index terms](#)

Networks have become a key component of the corporate infrastructure. Managing the networks, which often carry a diverse set of information ∥e.g. voice, data, video∥ over a diverse set of media ∥e.g. wire, cable, RF∥ with a mixture of owned and leased transmission assets that are often geographically distributed and run a diverse set of protocols, is a major challenge. One of the most promising techniques applies expert system approaches to the management of networks. Co ...

15 Fault detection with multiple observers

Clark Wang, Mischa Schwartz

February 1993 **IEEE/ACM Transactions on Networking (TON)**, Volume 1 Issue 1


Full text available:  pdf(854.36 KB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

16 Secure routing: Secure data transmission in mobile ad hoc networks

Panagiotis Papadimitratos, Zygmunt J. Haas

September 2003 **Proceedings of the 2003 ACM workshop on Wireless security**

Full text available:  pdf(1.18 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


The vision of nomadic computing with its ubiquitous access has stimulated much interest in the Mobile Ad Hoc Networking (MANET) technology. However, its proliferation strongly depends on the availability of security provisions, among other factors. In the open, collaborative MANET environment practically any node can maliciously or selfishly disrupt and deny communication of other nodes. In this paper, we present and evaluate the Secure Message Transmission (SMT) protocol, which safeguards the d ...

Keywords: MANET security, multi-path routing, secure message transmission, secure routing, secure routing protocol

17 Verisim: Formal analysis of network simulations

Karthikeyan Bhargavan, Carl A. Gunter, Moonjoo Kim, Insup Lee, Davor Obradovic, Oleg Sokolsky, Mahesh Viswanathan

August 2000 **ACM SIGSOFT Software Engineering Notes , Proceedings of the International Symposium on Software Testing and Analysis**, Volume 25 Issue 5

Full text available:  [pdf\(325.27 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Why are there so few successful "real-world" programming and testing tools based on academic research? This talk focuses on program analysis tools, and proposes a surprisingly simple explanation with interesting ramifications. For a tool aimed at developers or testers to be successful, people must use it - and must use it to help accomplish their existing tasks, rather than as an end in itself. If the tool does not help them get their job done, or the effort to learn and/or use th ...

18 File and storage systems: Preserving peer replicas by rate-limited sampled voting

Petros Maniatis, David S. H. Rosenthal, Mema Roussopoulos, Mary Baker, TJ Giuli, Yanto Muliadi

October 2003 **Proceedings of the nineteenth ACM symposium on Operating systems principles**

Full text available:  [pdf\(336.27 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The LOCKSS project has developed and deployed in a world-wide test a peer-to-peer system for preserving access to journals and other archival information published on the Web. It consists of a large number of independent, low-cost, persistent web caches that cooperate to detect and repair damage to their content by voting in "opinion polls." Based on this experience, we present a design for and simulations of a novel protocol for voting in systems of this kind. It incorporates rate limitation an ...

Keywords: digital preservation, rate limiting, replicated storage

19 Astrolabe: A robust and scalable technology for distributed system monitoring, management, and data mining

Robbert Van Renesse, Kenneth P. Birman, Werner Vogels

May 2003 **ACM Transactions on Computer Systems (TOCS)**, Volume 21 Issue 2

Full text available:  [pdf\(341.62 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Scalable management and self-organizational capabilities are emerging as central requirements for a generation of large-scale, highly dynamic, distributed applications. We have developed an entirely new distributed information management system called Astrolabe. Astrolabe collects large-scale system state, permitting rapid updates and providing on-the-fly attribute aggregation. This latter capability permits an application to locate a resource, and also offers a scalable way to track sys ...

Keywords: Aggregation, epidemic protocols, failure detection, gossip, membership, publish-subscribe, scalability

20 Session summaries from the 17th symposium on operating systems principle (SOSP'99)

Jay Lepreau, Eric Eide

April 2000 **ACM SIGOPS Operating Systems Review**, Volume 34 Issue 2

Full text available:  [pdf\(3.15 MB\)](#) Additional Information: [full citation](#), [index terms](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2003 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)



US Patent & Trademark Office

[Subscribe](#) (Full Service) [Register](#) (Limited Service, Free) [Login](#)

Search: ☒ The ACM Digital Library ☐ The Guide

(fsm and simulat* and network* and securit* and fault* and p

[SEARCH](#)

THE ACM DIGITAL LIBRARY

[Feedback](#) [Report a problem](#) [Satisf](#)

Terms used

fsm and **simulat** and **network** and **securit** and **fault** or **error** or **mistake** or **glitch** or **intrusion** and **predict** or **est**

Sort results by [relevance](#)

[Save results to a Binder](#)

[Try an Advanced Search](#)

Display results [expanded form](#)

[Search Tips](#)

Try this search in [The AC](#)

[Open results in a new window](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

R

1 [Illustrative risks to the public in the use of computer systems and related technology](#)

Peter G. Neumann

January 1996 **ACM SIGSOFT Software Engineering Notes**, Volume 21 Issue 1

Full text available: [pdf\(2.54 MB\)](#)

Additional Information: [full citation](#)

2 [Columns: Risks to the public in computers and related systems](#)

Peter G. Neumann

January 2001 **ACM SIGSOFT Software Engineering Notes**, Volume 26 Issue 1

Full text available: [pdf\(3.24 MB\)](#)

Additional Information: [full citation](#)

3 [Modeling for text compression](#)

Timothy Bell, Ian H. Witten, John G. Cleary

December 1989 **ACM Computing Surveys (CSUR)**, Volume 21 Issue 4

Full text available: [pdf\(3.54 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index te](#)

The best schemes for text compression use large models to help them predict which characters will actual next characters are coded with respect to the prediction, resulting in compression of inform best formed adaptively, based on the text seen so far. This paper surveys successful strategies for modeling that are suitable for use in practical text compression systems. The strategies fall into th finite-context modeling, i ...

4 [Power minimization in IC design: principles and applications](#)

Massoud Pedram

January 1996 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volum

Full text available: [pdf\(550.02 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index te](#)

Low power has emerged as a principal theme in today's electronics industry. The need for low pow major paradigm shift in which power dissipation is as important as performance and area. This art in-depth survey of CAD methodologies and techniques for designing low power digital CMOS circui and describes the many issues facing designers at architectural, logical, and physical levels of desi reviews some of the techniques and tool ...


Keywords: CMOS circuits, adiabatic circuits, computer-aided design of VLSI, dynamic power dissi

delay product, gated clocks, layout, low power layout, low power synthesis, lower-power design, p estimation, power management, power minimization and management, probabilistic analysis, silico technology, statistical sampling, switched capacitance, switching activity, symbolic simulation, syn design

5 Columns: Risks to the public in computers and related systems

Peter G. Neumann

May 2000 **ACM SIGSOFT Software Engineering Notes**, Volume 25 Issue 3

Full text available:  pdf(1.11 MB) Additional Information: [full citation](#)

6 Risks to the public in computers and related systems

Peter G. Neumann

January 1999 **ACM SIGSOFT Software Engineering Notes**, Volume 24 Issue 1

Full text available:  pdf(625.70 KB) Additional Information: [full citation](#), [index terms](#)

7 Fast detection of communication patterns in distributed executions

Thomas Kunz, Michiel F. H. Seuren

November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on C research**

Full text available:  pdf(4.21 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Understanding distributed applications is a tedious and difficult task. Visualizations based on proce are often used to obtain a better understanding of the execution of the application. The visualizatio Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often do not provide the user with the desired overview of the application. In our experience, such tools occurrences of non-trivial commun ...

8 Terrain database interoperability issues in training with distributed interactive simulation

Guy A. Schiavone, S. Sureshchandran, Kenneth C. Hardis

July 1997 **ACM Transactions on Modeling and Computer Simulation (TOMACS)**, Volume 7 Issu

Full text available:  pdf(443.34 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citings](#), [index te](#)

In Distributed Interactive Simulation (DIS), each participating node is responsible for maintaining the synthetic environment. Problems may arise if significant inconsistencies are allowed to exist be separate world views, resulting in unrealistic simulation results or negative training, and a corresp degradation of interoperability in a DIS simulation exercise. In the DIS community, this is known a terrain database (TDB) correlation problem. ...

Keywords: distributed interactive simulation, terrain databases

9 Local error recovery in SRM: comparison of two approaches



Ching-Gung Liu, Deborah Estrin, Scott Shenker, Lixia Zhang

December 1998 **IEEE/ACM Transactions on Networking (TON)**, Volume 6 Issue 6

Full text available:  pdf(539.49 KB) Additional Information: [full citation](#), [references](#), [index terms](#)

10 Computing curricula 2001

September 2001 **Journal on Educational Resources in Computing (JERIC)**

Full text available:  pdf(613.63 KB)  html
(2.78 KB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

11 Power estimation techniques for integrated circuits

Farid N. Najm

December 1995 **Proceedings of the 1995 IEEE/ACM international conference on Computer-aid**

Full text available:  pdf(218.32 KB) 
[Publisher Site](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index te](#)

With the advent of portable and high-density microelectronic devices, the power dissipation of very integrated (VLSI) circuits is becoming a critical concern. Accurate and efficient power estimation d phase is required in order to meet the power specifications without a costly redesign process. Rece power estimation techniques have been proposed, most of which are based on: 1) the use of simp models, and 2) modeling the long-term behavior ...

Keywords: power estimation VLSI circuit survey tutorial probability statistics

12 IDMaps: a global internet host distance estimation service

Paul Francis, Sugih Jamin, Cheng Jin, Yixin Jin, Danny Raz, Yuval Shavitt, Lixia Zhang

October 2001 **IEEE/ACM Transactions on Networking (TON)**, Volume 9 Issue 5

Full text available:  pdf(267.64 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index te](#)


There is an increasing need to quickly and efficiently learn network distances, in terms of metrics s bandwidth, between Internet hosts. For example, Internet content providers often place data and throughout the Internet to improve access latency for clients, and it is necessary to direct clients t mirrors based on some distance metric in order to realize the benefit of mirrors. We suggest a sca wide architecture, called IDMaps, which m ...

Keywords: Distributed algorithms, modeling, network service, scalability

13 Performance: On estimating end-to-end network path properties

Mark Allman, Vern Paxson

April 2001 **ACM SIGCOMM Computer Communication Review**, Volume 31 Issue 2 supplement

Full text available:  pdf(3.07 MB)


Additional Information: [full citation](#), [abstract](#), [references](#)

The more information about current network conditions available to a transport protocol, the more use the network to transfer its data. In networks such as the Internet, the transport protocol must own estimates of network properties based on measurements performed by the connection endpoi two basic transport estimation problems: determining the setting of the retransmission timer (RTO protocol, and estimating the bandwidth availa ...

14 Network interactions: Experiences with worm propagation simulations

Arno Wagner, Thomas Dübendorfer, Bernhard Plattner, Roman Hiestand

October 2003 **Proceedings of the 2003 ACM workshop on Rapid Malcode**

Full text available:  pdf(220.07 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


Fast Internet worms are a relatively new threat to Internet infrastructure and hosts. We discuss m possibilities to study the behaviour of such worms and degrees of freedom that worm writers have study of fast worms we have designed a simulator. We describe the design of this simulator and di experiences we have made with it and compare observation of past worms with simulated behavior feature of the simulator is that the Internet mod ...

Keywords: bandwidth, internet worms, latency, simulation

15 Low power scalable encryption for wireless systems

James Goodman, Anantha P. Chandrakasan

January 1998 **Wireless Networks**, Volume 4 Issue 1

Full text available:  pdf(7.39 MB)


Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Secure transmission of multimedia information (e.g., voice, video, data, etc.) is critical in many wireless applications. Wireless transmission imposes constraints not found in typical wired systems such as power consumption, tolerance to high bit error rates, and scalability. A variety of low power techniques have been developed to reduce the power of several encryption algorithms. One key idea involves exploiting computation requirements to dynamically vary the ...

16 On estimating end-to-end network path properties

Mark Allman, Vern Paxson

August 1999 **ACM SIGCOMM Computer Communication Review , Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication**, Issue 4

Full text available:  pdf(1.75 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The more information about current network conditions available to a transport protocol, the more it can use the network to transfer its data. In networks such as the Internet, the transport protocol must maintain its own estimates of network properties based on measurements performed by the connection endpoints. There are two basic transport estimation problems: determining the setting of the retransmission timer (RTO) and estimating the bandwidth available ...

17 Practical byzantine fault tolerance and proactive recovery

Miguel Castro, Barbara Liskov

November 2002 **ACM Transactions on Computer Systems (TOCS)**, Volume 20 Issue 4

Full text available:  pdf(1.63 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#), [reviews](#)


Our growing reliance on online services accessible on the Internet demands highly available systems that provide correct service without interruptions. Software bugs, operator mistakes, and malicious attacks are common causes of service interruptions and they can cause arbitrary behavior, that is, Byzantine faults. This article describes a replication algorithm, BFT, that can be used to build highly available systems that tolerate Byzantine faults. It can be used in practice to implement replicated state machine ...

Keywords: Byzantine fault tolerance, asynchronous systems, proactive recovery, state machine replication, transfer

18 Exact and approximate methods for calculating signal and transition probabilities in FSMs

Chi-Ying Tsui, Massoud Pedram, Alvin M. Despain

June 1994 **Proceedings of the 31st annual conference on Design automation conference**

Full text available:  pdf(232.30 KB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

19 Creating trading networks of digital archives

Brian Cooper, Hector Garcia

January 2001 **Proceedings of the first ACM/IEEE-CS joint conference on Digital libraries**

Full text available:  pdf(785.50 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Digital archives can best survive failures if they have made several copies of their collections at remote sites. In this paper, we discuss how autonomous sites can cooperate to provide preservation by trading data ...

the decisions that an archive must make when forming trading networks, such as the amount of st provide and the best number of partner sites. We also deal with the fact that some sites may be m others. Experimental results from a data t ...

Keywords: data trading, digital archiving, fault tolerance, preservation, replication

20 Simulating population and employment change for U.S. metropolitan and rural areas

Peter M. Allaman

December 1978 **Proceedings of the 10th conference on Winter simulation - Volume 2**

Full text available:  pdf(878.74 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper reports on a computer simulation model of migration and employment change in 315 a together constitute the contiguous United States. In the process of constructing this model, an ext of 1960 and 1970 social and economic data was assembled at the county level from the Census of Housing and procedures were developed for aggregating these data to more meaningful functional counties. This provided measures of levels of activities i ...

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2003 ACM, In
[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Play](#)

[IEEE HOME](#) | [SEARCH IEEE](#) | [SHOP](#) | [WEB ACCOUNT](#) | [CONTACT IEEE](#)[Membership](#) [Publications/Services](#) [Standards](#) [Conferences](#) [Careers/Jobs](#)**IEEE Xplore®**
RELEASE 1.5Welcome
United States Patent and Trademark Office[Help](#) [FAQ](#) [Terms](#) [IEEE Peer Review](#)[Quick Links](#)[» Se](#)

Welcome to IEEE Xplore®

Your search matched **[0]** of **[989514]** documents.

- ☐ Home
- ☐ What Can I Access?
- ☐ Log-out

Tables of Contents

- ☐ Journals & Magazines
- ☐ Conference Proceedings
- ☐ Standards

Search

- ☐ By Author
- ☐ Basic
- ☐ Advanced

Member Services

- ☐ Join IEEE
- ☐ Establish IEEE Web Account
- ☐ Access the IEEE Member Digital Library

[Print Format](#)

You may refine your search by editing the current search expression or entering a new one the text box. Then click search Again.

[Search Again](#)**OR**

Use your browser's back button to return to your original search page.

Results:

No documents matched your query.

[Home](#) | [Log-out](#) | [Journals](#) | [Conference Proceedings](#) | [Standards](#) | [Search by Author](#) | [Basic Search](#) | [Advanced Search](#)
[Join IEEE](#) | [Web Account](#) | [New this week](#) | [OPAC Linking Information](#) | [Your Feedback](#) | [Technical Support](#) | [Email Alerting](#)
[No Robots Please](#) | [Release Notes](#) | [IEEE Online Publications](#) | [Help](#) | [FAQ](#) | [Terms](#) | [Back to Top](#)

Copyright © 2003 IEEE — All rights reserved

IEEE HOME | SEARCH IEEE | SHOP | WEB ACCOUNT | CONTACT IEEE



Membership Publications/Services Standards Conferences Careers/Jobs

IEEE Xplore®
 RELEASE 1.5

 Welcome
 United States Patent and Trademark Office

[Help](#) [FAQ](#) [Terms](#) [IEEE Peer Review](#)
[Quick Links](#)

» Adva

Welcome to IEEE Xplore®

- ☐ Home
- ☐ What Can I Access?
- ☐ Log-out

Tables of Contents

- ☐ Journals & Magazines
- ☐ Conference Proceedings
- ☐ Standards

Search

- ☐ By Author
- ☐ Basic
- ☐ Advanced

Member Services

- ☐ Join IEEE
- ☐ Establish IEEE Web Account
- ☐ Access the IEEE Member Digital Library

1) Enter a single keyword, phrase, or Boolean expression.
 Example: acoustic imaging (means the phrase acoustic imaging plus any stem variations)

2) Limit your search by using search operators and field codes, if desired.

Example: optical <and> (fiber <or> fibre) <in> ti

3) Limit the results by selecting Search Options.

4) Click Search. See [Search Examples](#)

```
(((FSM or IFSM or (finite
<near/1> state <near/1>
machine)) and simulat*) and
network) and securit*)) and
```

Start Search

Clear

Note: This function returns plural and suffixed forms of the keyword(s).

Search operators: <and> <or> <not> <in> [More](#)

Field codes: au (author), ti (title), ab (abstract), jn (publication name), de (index term) [More](#)

Search Options:

Select publication types:

- ☒ IEEE Journals
- ☒ IEE Journals
- ☒ IEEE Conference proceed
- ☒ IEE Conference proceedin
- ☒ IEEE Standards

Select years to search:

 From year: to

Organize search results by

 Sort by:

 In: order

 List Results per page

[Home](#) | [Log-out](#) | [Journals](#) | [Conference Proceedings](#) | [Standards](#) | [Search by Author](#) | [Basic Search](#) | [Advanced Search](#)
[Join IEEE](#) | [Web Account](#) | [New this week](#) | [OPAC Linking Information](#) | [Your Feedback](#) | [Technical Support](#) | [Email Alerting](#) | [No Robots Please](#) |
[Release Notes](#) | [IEEE Online Publications](#) | [Help](#) | [FAQ](#) | [Terms](#) | [Back to Top](#)

Copyright © 2003 IEEE — All rights reserved



[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

Search: ☒ The ACM Digital Library ☐ The Guide

THE ACM DIGITAL LIBRARY

[Feedback](#) [Rep](#)

Terms used

finite near/1 state near/1 machine and monitor and simulat and network and securit and stochastic and cont

Sort results by

Display results

[Save results to a Binder](#)

[Search Tips](#)

☒ [Open results in a new window](#)

Try an
Try th

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

1 [Computing curricula 2001](#)

September 2001 **Journal on Educational Resources in Computing (JERIC)**

Full text available: [pdf\(613.63 KB\)](#) [html\(2.78 KB\)](#)

Additional Information: [full citation](#), [references](#), [citing](#), [inde](#)

2 [Fast detection of communication patterns in distributed executions](#)

Thomas Kunz, Michiel F. H. Seuren

November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on C**

Full text available: [pdf\(4.21 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [ind](#)

Understanding distributed applications is a tedious and difficult task. Visualizations based on proce better understanding of the execution of the application. The visualization tool we use is Poet, an e Waterloo. However, these diagrams are often very complex and do not provide the user with the d experience, such tools display repeated occurrences of non-trivial commun ...

3 [Magic Functions: In Memoriam: Bernard M. Dwork 1923--1998](#)

Cynthia Dwork, Moni Naor, Omer Reingold, Larry Stockmeyer

November 2003 **Journal of the ACM (JACM)**, Volume 50 Issue 6

Full text available: [pdf\(708.05 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [ind](#)

We prove that three apparently unrelated fundamental problems in distributed computing, cryptog the same problem. These three problems and brief descriptions of them follow. (1) *The selective d* commitments to a collection of messages, and the adversary can ask for some subset of the comm seeing the decommitments to these open plaintexts allows the adversary t ...

Keywords: Digital signature, Fiat-Shamir methodology, interactive argument, interactive proof sy zero knowledge

4 [Enhanced operational semantics: a tool for describing and analyzing concurrent systems](#)

Pierpaolo Degano, Corrado Priami

June 2001 **ACM Computing Surveys (CSUR)**, Volume 33 Issue 2

Full text available: [pdf\(355.24 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [cit](#)

This article surveys the definition and application of an enhancement of structural operational sem

also addresses issues of distribution and mobility of code. The focus is on how enriching the labels trees is sufficient to derive qualitative and quantitative information on the systems in hand simply concrete model.

Keywords: parametric models, process algebra

5 On randomization in sequential and distributed algorithms

Rajiv Gupta, Scott A. Smolka, Shaji Bhaskar

March 1994 **ACM Computing Surveys (CSUR)**, Volume 26 Issue 1

Full text available:  pdf(8.01 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [cit](#)


Probabilistic, or randomized, algorithms are fast becoming as commonplace as conventional deterministic techniques that have been widely used in the design of randomized algorithms. These techniques are both sequential and distributed— that span a wide range of applications, including: primality testing, interactive probabilistic proofs ...

Keywords: Byzantine agreement, CSP, analysis of algorithms, computational complexity, dining table graph isomorphism, hashing, interactive probabilistic proof systems, leader election, message routing, hashing, primality testing, probabilistic techniques, randomized or probabilistic algorithms, random tournaments, universal hashing

6 Real-time estimation of the parameters of long-range dependence

Matthew Roughan, Darryl Veitch, Patrice Abry

August 2000 **IEEE/ACM Transactions on Networking (TON)**, Volume 8 Issue 4

Full text available:  pdf(237.43 KB)


Additional Information: [full citation](#), [references](#), [citations](#), [index term](#)

Keywords: Hurst parameter, estimation, fractal, long-range dependence, on-line, real-time, self-s

7 Programming languages and systems for prototyping concurrent applications

Wilhelm Hasselbring

March 2000 **ACM Computing Surveys (CSUR)**, Volume 32 Issue 1

Full text available:  pdf(559.78 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [ind](#)


Concurrent programming is conceptually harder to undertake and to understand than sequential programming. To manage the coexistence and coordination of multiple concurrent activities, to alleviate this task several programming languages and systems have been developed. For some high-level programming approaches, prototyping is a central goal. Prototyping is used to explore the ...

Keywords: concurrency, distribution, parallelism, rapid prototyping, very high-level languages

8 Bimodal multicast

Kenneth P. Birman, Mark Hayden, Ozgur Ozkasap, Zhen Xiao, Mihai Budiu, Yaron Minsky

May 1999 **ACM Transactions on Computer Systems (TOCS)**, Volume 17 Issue 2

Full text available:  pdf(302.06 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [cit](#)

There are many methods for making a multicast protocol "reliable." At one end of the spectrum, a protocol guarantees, such as all-or-nothing delivery, delivery ordering, and perhaps additional properties such as other are protocols that use local repair to overcome transient packet loss in the network, offering work has treated stability ...

9 Astrolabe: A robust and scalable technology for distributed system monitoring, management

Robbert Van Renesse, Kenneth P. Birman, Werner Vogels

May 2003

ACM Transactions on Computer Systems (TOCS), Volume 21 Issue 2

Full text available:  pdf(341.62 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [ind](#)

Scalable management and self-organizational capabilities are emerging as central requirements for distributed applications. We have developed an entirely new distributed information management scale system state, permitting rapid updates and providing on-the-fly attribute aggregation. This is a resource, and also offers a scalable way to track sys ...

Keywords: Aggregation, epidemic protocols, failure detection, gossip, membership, publish-subsc

10 Process migration

September 2000 **ACM Computing Surveys (CSUR)**, Volume 32 Issue 3

Full text available:  pdf(1.24 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [cit](#)


Process migration is the act of transferring a process between two machines. It enables dynamic load administration, and data access locality. Despite these goals and ongoing research efforts, migration is increasing deployment of distributed systems in general, and distributed operating systems in particular, attracting attention in both research and product development. As hi ...

Keywords: distributed operating systems, distributed systems, load distribution, process migration

11 The proposed new Computing Reviews classification scheme

Anthony Ralston

July 1981 **Communications of the ACM**, Volume 24 Issue 7


Full text available:  pdf(972.02 KB)

Additional Information: [full citation](#), [citations](#), [index terms](#)

12 IS '97: model curriculum and guidelines for undergraduate degree programs in information systems

Gordon B. Davis, John T. Gorgone, J. Daniel Cougar, David L. Feinstein, Herbert E. Longenecker

December 1997 **ACM SIGMIS Database, Guidelines for undergraduate degree programs on MIS**
undergraduate degree programs in information systems, Volume 28 Issue 1

Full text available:  pdf(7.24 MB)

Additional Information: [full citation](#), [citations](#)

13 Network interactions: Simulating realistic network worm traffic for worm warning system design

Michael Liljenstam, David M. Nicol, Vincent H. Berk, Robert S. Gray

October 2003 **Proceedings of the 2003 ACM workshop on Rapid Malcode**

Full text available:  pdf(308.41 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [ind](#)

Reproducing the effects of large-scale worm attacks in a laboratory setup in a realistic and reproducible development of worm detection and defense systems. In this paper, we describe a worm simulation that reproduces the large-scale spread dynamics of a worm and many aspects of its detailed effects on the network, including realistic scan rates on realistic IP address spaces and selectively model local dynamics ...

Keywords: code red, network modeling and simulation, network security, slammer, worm detection

The new (1982) Computing Reviews classification system—final version

Jean E. Sammet, Anthony Ralston

January 1982 **Communications of the ACM**, Volume 25 Issue 1

Full text available:  [pdf\(731.04 KB\)](#)

Additional Information: [full citation](#), [citations](#), [index terms](#)

15 Superfast parallel discrete event simulations

Albert G. Greenberg, Boris D. Lubachevsky, Isi Mitrani

April 1996 **ACM Transactions on Modeling and Computer Simulation (TOMACS)**, Volume 6

Full text available:  [pdf\(421.63 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


Nonconventional parallel simulations methods are presented, wherein speed-ups are not limited by methods capitalize on Chandy and Sherman's space-time relaxation paradigm, and incorporate fast attention is paid to implementing these algorithms on currently available massively parallel SIMD computers for open and closed queuing networks and for the slotted ALOHA ...

Keywords: fixed-point computations, massively parallel computations, recurrences, relaxation, simulation

16 A Survey of Some Theoretical Aspects of Multiprocessing

J. L. Baer

January 1973 **ACM Computing Surveys (CSUR)**, Volume 5 Issue 1


Full text available:  [pdf\(4.05 MB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

17 Session summaries from the 17th symposium on operating systems principles (SOSP'99)

Jay Lepreau, Eric Eide

April 2000 **ACM SIGOPS Operating Systems Review**, Volume 34 Issue 2


Full text available:  [pdf\(3.15 MB\)](#)

Additional Information: [full citation](#), [index terms](#)

18 Safely executing untrusted code: Model-carrying code: a practical approach for safe execution

R. Sekar, V.N. Venkatakrishnan, Samik Basu, Sandeep Bhatkar, Daniel C. DuVarney

October 2003 **Proceedings of the nineteenth ACM symposium on Operating systems principles**

Full text available:  [pdf\(301.30 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper presents a new approach called *model-carrying code* (MCC) for safe execution of untrusted code. The code comes equipped with a concise high-level model of its security-relevant behavior. The model is used to enforce high-level security policies and low-level binary code, thereby enabling analyses which would otherwise require a fully automated verification procedure to determine if the code is safe ...

Keywords: mobile code security, policy enforcement, sand-boxing, security policies

19 Programming languages for distributed computing systems

Henri E. Bal, Jennifer G. Steiner, Andrew S. Tanenbaum

September 1989 **ACM Computing Surveys (CSUR)**, Volume 21 Issue 3

Full text available:  [pdf\(6.50 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

When distributed systems first appeared, they were programmed in traditional sequential language procedures for sending and receiving messages. As distributed applications became more common, the message-passing approach became less satisfactory. Researchers all over the world began designing new programming languages for distributed systems ...

distributed applications. These languages and their history, their underlying pr ...

20 Information survivability control systems

Kevin Sullivan, John C. Knight, Xing Du, Steve Geist

May 1999 **Proceedings of the 21st international conference on Software engineering**

Full text available:  [pdf\(1.23 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Keywords: architecture economics, control, infrastructure survivability

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright
[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Play](#)



US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)Search: ☒ The ACM Digital Library ☐ The Guide

keyword:"worm detection systems"

SEARCH

THE ACM DIGITAL LIBRARY

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)Terms used **worm detection systems**

Found 1 of 124,098

Sort results
by

relevance

[Save results to a Binder](#)Try an [Advanced Search](#)Display
results

expanded form

[Search Tips](#)Try this search in [The ACM Guide](#)☐ Open results in a new
window

Results 1 - 1 of 1

Relevance scale ☐☐☐☐☐**1 [Network interactions: Simulating realistic network worm traffic for worm warning system design and testing](#)**

Michael Liljenstam, David M. Nicol, Vincent H. Berk, Robert S. Gray

October 2003 **Proceedings of the 2003 ACM workshop on Rapid Malcode**Full text available: [pdf\(308.41 KB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Reproducing the effects of large-scale worm attacks in a laboratory setup in a realistic and reproducible manner is an important issue for the development of worm detection and defense systems. In this paper, we describe a worm simulation model we are developing to accurately model the large-scale spread dynamics of a worm and many aspects of its detailed effects on the network. We can model slow or fast worms with realistic scan rates on realistic IP address spaces and selectively model local d ...

Keywords: code red, network modeling and simulation, network security, slammer, worm detection systems, worms

Results 1 - 1 of 1

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2003 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)Useful downloads: [Adobe Acrobat](#) [QuickTime](#) [Windows Media Player](#) [Real Player](#)



US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

CCS:"Network monitoring"

SEARCH

THE ACM DIGITAL LIBRARY

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Terms used Network monitoring

Found 139 of 124,098

Sort results
by

relevance

[Save results to a Binder](#)Try an [Advanced Search](#)Display
results

expanded form

[Search Tips](#)Try this search in [The ACM Guide](#)
☐ Open results in a new
window

Results 1 - 20 of 139

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [next](#)Relevance scale ☐ ☐ ☐ ☐ ☐1 Enforcing model network citizenship by remote administration

Prasun Gupta, Mahmoud Pegah

September 2003 **Proceedings of the 31st annual ACM SIGUCCS conference on User services**Full text available: [pdf\(493.86 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Higher education institutions have been striving to improve services and keep pace with new technologies. In a Higher education environment, the users utilize the available computing resources 24 hours a day 7 days a week. Although we cannot have the 24 by 7 uptime guaranteed, due to issues like budget constraints, we could certainly reduce the downtime by deploying a cost effective open source network monitoring solution such as Big Brother. Commercially available network management systems are ...

Keywords: RMON, SNMP, big brother, monitoring, network management, network monitoring, system monitoring, system reliability

2 Deployment and testbeds: Enhancement of a WLAN-based internet service in Korea
 Youngkyu Choi, Jeongyeup Paek, Sunghyun Choi, Go Woon Lee, Jae Hwan Lee, Hanwook Jung
 September 2003 **Proceedings of the 1st ACM international workshop on Wireless mobile applications and services on WLAN hotspots**
Full text available: [pdf\(774.23 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

A wireless LAN (WLAN)-based Internet service, called NESPOT, of Korea Telecom (KT), the biggest telecommunication and Internet service company in Korea, has been operational since early 2002. As the numbers of subscribers and deployed access points (APs) increase, KT has been endeavoring to improve its service quality as well as the network management. In this paper, we introduce a joint effort between Seoul National University (SNU) and KT to achieve it. We have been addressing two major issues ...

Keywords: IEEE 802.11, LAN, hotspot service, wireless internet service provider (WISP)

3 A hierarchical multicast monitoring scheme

Joerg Walz, Brian Neil Levine

November 2000 **Proceedings of NGC 2000 on Networked group communication**Full text available: [pdf\(1.29 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Deployment of multicast routing services in corporate networks and Internet Service Providers is still tentative. Among other problems, there is a lack of monitoring and management tools and systems. Previous work in multicast management has failed to address the scalability problem present in multicast fault isolation and reporting. We propose a hierarchical, passive monitoring scheme, HPMM, that relies on a series of pre-deployed, self-organized monitoring daemons. With HPMM, fault message ...

4 Controlling unresponsive connections in an active network architecture

Niraj Prabhavalkar, Manish Parashar

July 2003 **International Journal of Network Management**, Volume 13 Issue 4

Full text available:  [pdf\(195.85 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper presents the design, implementation and evaluation of Limiting Greedy Connections (LGC), an active mechanism for controlling unresponsive connections and minimizing the degradation in network performance caused by bandwidth-greedy applications. The primary objectives of the LGC mechanism are to limit the impact of greedy connections on a congested node, to keep a loose upper bound on the packet queue occupancy at the intermediate nodes of the network and to minimize packet loss. The L ...

5 Development of SNMP-XML translator and gateway for XML-based integrated network management

Jeong-Hyuk Yoon, Hong-Taek Ju, James W. Hong

July 2003 **International Journal of Network Management**, Volume 13 Issue 4

Full text available:  [pdf\(251.82 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The research objective of our work is to develop a SNMP MIB to XML translation algorithm and to implement an SNMP-XML gateway using this algorithm. The gateway is used to transfer management information between an XML-based manager and SNMP-based agents. SNMP is widely used for Internet management, but SNMP is insufficient to manage continuously expanding networks because of constraints in scalability and efficiency. XML based network management architectures are newly proposed as alternatives t ...

6 Clustering intrusion detection alarms to support root cause analysis

Klaus Julisch

November 2003 **ACM Transactions on Information and System Security (TISSEC)**, Volume 6 Issue 4

Full text available:  [pdf\(285.72 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

It is a well-known problem that intrusion detection systems overload their human operators by triggering thousands of alarms per day. This paper presents a new approach for handling intrusion detection alarms more efficiently. Central to this approach is the notion that each alarm occurs for a reason, which is referred to as the alarm's *root causes*. This paper observes that a few dozens of rather persistent root causes generally account for over 90&percent; of the alarms that an intrusion ...

Keywords: Intrusion detection, cluster analysis, data mining, false positives, root cause analysis

7 Measuring and characterizing end-to-end Internet service performance

Ludmila Cherkasova, Yun Fu, Wenting Tang, Amin Vahdat

November 2003 **ACM Transactions on Internet Technology (TOIT)**, Volume 3 Issue 4

Full text available:  [pdf\(1.46 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Fundamental to the design of reliable, high-performance network services is an understanding of the performance characteristics of the service as perceived by the client

population as a whole. Understanding and measuring such end-to-end service performance is a challenging task. Current techniques include periodic sampling of service characteristics from strategic locations in the network and instrumenting Web pages with code that reports client-perceived latency back to a performance server. Li ...

Keywords: End-to-end service performance, QoS, network packet traces, passive monitoring, reconstruction of web page composition, web site performance

8 The effect of uncertain time-variant delays in ATM networks with explicit rate feedback: a control theoretic approach ☐

Mihail L. Sichitiu, Peter H. Bauer, Kamal Premaratne

August 2003 **IEEE/ACM Transactions on Networking (TON)**, Volume 11 Issue 4

Full text available:  pdf(532.25 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

A new, more realistic model for the available bit rate traffic class in ATM network congestion control with explicit rate feedback is introduced and analyzed. This model is based on recent results by Ekanayake regarding discrete time models for time-variant delays. The discrete time model takes into account the effect of time-variant buffer occupancy levels of ATM switches, thus treating the case of time-variant delays between a single congested node and the connected sources. For highly dynamic ...

9 Concurrent fault detection for a multiple-plane packet switch ☐

Roberto Rojas-Cessa, Eiji Oki, H. Jonathan Chao

August 2003 **IEEE/ACM Transactions on Networking (TON)**, Volume 11 Issue 4

Full text available:  pdf(711.66 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

In high-speed and high-capacity packet switches, system reliability is critical to avoid loss of huge amounts of information and retransmission of traffic. We propose a series of concurrent fault-detection mechanisms for a multiple-plane crossbar-based packet switch. Our switch model, called the $m+z$ model, has m active planes and z spare planes. This switch has distributed arbiters on each plane. The spare planes, used for substitution of faulty active ones, are also ...

Keywords: concurrent testing, fault detection, packet switch, parallel planes, single fault

10 End-to-end rate-based congestion control: convergence properties and scalability analysis ☐

Dmitri Loguinov, Hayder Radha

August 2003 **IEEE/ACM Transactions on Networking (TON)**, Volume 11 Issue 4

Full text available:  pdf(606.83 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

In this paper, we study several properties of binary-feedback congestion control in rate-based applications. We first derive necessary conditions for generic binary-feedback congestion control to converge to fairness monotonically (which guarantees asymptotic stability of the fairness point) and show that AIMD is the *only* TCP-friendly binomial control with monotonic convergence to fairness. We then study steady-state behavior of binomial controls with n competing flows on a single ...

Keywords: MPEG-4, binomial algorithms, congestion control, multimedia streaming, packet loss scalability

11 ☐

End-to-end available bandwidth: measurement methodology, dynamics, and relation

with TCP throughput

Manish Jain, Constantinos Dovrolis

August 2003 **IEEE/ACM Transactions on Networking (TON)**, Volume 11 Issue 4Full text available:  pdf(934.74 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The available bandwidth (avail-bw) in a network path is of major importance in congestion control, streaming applications, quality-of-service verification, server selection, and overlay networks. We describe an end-to-end methodology, called self-loading periodic streams (SLoPS), for measuring avail-bw. The basic idea in SLoPS is that the one-way delays of a periodic packet stream show an increasing trend when the stream's rate is higher than the avail-bw. We implemented SLoPS in a tool called < ...

Keywords: active probing, bottleneck bandwidth, bulk transfer capacity, network capacity, packet pair dispersion

12 Characterization: Network performance monitoring at small time scales 

Konstantina Papagiannaki, Rene Cruz, Christophe Diot

October 2003 **Proceedings of the conference on Internet measurement conference**Full text available:  pdf(193.77 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

SNMP statistics are usually collected over intervals of 5 minutes and correspond to average activity of IP links and network elements for the duration of the interval. Nevertheless, reports of traffic performance across periods of minutes can mask out performance degradation due to short-lived events, such as micro-congestion episodes, that manifest themselves at smaller time scales. In this paper we perform a measurement study of packet traces collected inside the Sprint IP network to identify ...

Keywords: congestion detection, internet measurement, performance monitoring

13 Approximations: Traffic engineering with estimated traffic matrices 

Matthew Roughan, Mikkell Thorup, Yin Zhang

October 2003 **Proceedings of the conference on Internet measurement conference**Full text available:  pdf(248.12 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Traffic engineering and traffic matrix estimation are often treated as separate fields, even though one of the major applications for a traffic matrix is traffic engineering. In cases where a traffic matrix cannot be measured directly, it may still be estimated from indirect data (such as link measurements), but these estimates contain errors. Yet little thought has been given to the effects of inexact traffic estimates on traffic engineering. In this paper we consider how well traffic engineeri ...

Keywords: MPLS, OSPF, SNMP, traffic engineering, traffic matrix estimation

14 Approximations: Sketch-based change detection: methods, evaluation, and applications 

Balachander Krishnamurthy, Subhabrata Sen, Yin Zhang, Yan Chen

October 2003 **Proceedings of the conference on Internet measurement conference**Full text available:  pdf(309.23 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Traffic anomalies such as failures and attacks are commonplace in today's network, and identifying them rapidly and accurately is critical for large network operators. The detection typically treats the traffic as a collection of flows that need to be examined for significant changes in traffic pattern (eg, volume, number of connections). However, as link speeds and

the number of flows increase, keeping per-flow state is either too expensive or too slow. We propose building compact summaries of ...

Keywords: change detection, data stream computation, forecasting, network anomaly detection, sketch, time series analysis

15 Approximations: Inverting sampled traffic

Nicolas Hohn, Darryl Veitch

October 2003 **Proceedings of the conference on Internet measurement conference**

Full text available:  [pdf\(308.04 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Routers have the ability to output statistics about packets and flows of packets that traverse them. Since however the generation of detailed traffic statistics does not scale well with link speed, increasingly routers and measurement boxes implement sampling strategies at the packet level. In this paper we study both theoretically and practically what information about the original traffic can be inferred when sampling, or 'thinning', is performed at the packet level. While basic packet level c ...

Keywords: Poisson cluster process, TCP flows, internet data, long range dependence, sampling, thinning, traffic modeling, transform inversion

16 Tomography: Tomography-based overlay network monitoring

Yan Chen, David Bindel, Randy H. Katz

October 2003 **Proceedings of the conference on Internet measurement conference**

Full text available:  [pdf\(259.22 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Overlay network monitoring enables distributed Internet applications to detect and recover from path outages and periods of degraded performance within seconds. For an overlay network with n end hosts, existing systems either require $O(n^2)$ measurements, and thus lack scalability, or can only estimate the latency but not congestion or failures. Unlike other network tomography systems, we characterize end-to-end losses (this extends to any additive metrics, including latency) rather than ...

Keywords: network measurement and monitoring, network tomography, numerical linear algebra, overlay networks

17 Tomography: Simple network performance tomography

Nick Duffield

October 2003 **Proceedings of the conference on Internet measurement conference**

Full text available:  [pdf\(177.77 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

In network performance tomography, characteristics of the network interior are inferred by correlating end-to-end measurements. In much previous work, the presence of correlations must be arranged at the packet level, e.g., using multicast probes or unicast emulations of them. This carries costs in deployment and limits coverage. However, it is difficult to determine performance characteristics without correlations. Some recent work has had success in reaching a lesser goal---identifying the loss ...

Keywords: correlation, estimation, inference, networks, performance

18 Algorithms: Predicting resource usage and estimation accuracy in an IP flow measurement collection infrastructure

Nick Duffield, Carsten Lund

October 2003 **Proceedings of the conference on Internet measurement conference**

Full text available:  pdf(362.83 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper describes a measurement infrastructure used to collect detailed IP traffic measurements from an IP backbone. Usage, i.e, bytes transmitted, is determined from raw NetFlow records generated by the backbone routers. The amount of raw data is immense. Two types of data sampling in order to manage data volumes: (i) (packet) sampled NetFlow in the routers; (ii) size-dependent sampling of NetFlow records. Furthermore, dropping of NetFlow records in transmission can be regarded as an uncontr ...

Keywords: bandwidth, estimation, sampling, variance

19 Algorithms: Identifying frequent items in sliding windows over on-line packet streams 

Lukasz Golab, David DeHaan, Erik D. Demaine, Alejandro Lopez-Ortiz, J. Ian Munro

October 2003 **Proceedings of the conference on Internet measurement conference**

Full text available:  pdf(211.86 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Internet traffic patterns are believed to obey the power law, implying that most of the bandwidth is consumed by a small set of heavy users. Hence, queries that return a list of frequently occurring items are important in the analysis of real-time Internet packet streams. While several results exist for computing frequent item queries using limited memory in the infinite stream model, in this paper we consider the limited-memory sliding window model. This model maintains the last N items that ...

Keywords: frequent item queries, internet traffic monitoring, on-line stream analysis, sliding windows

20 Algorithms: Space-code bloom filter for efficient traffic flow measurement 

Abhishek Kumar, Li Li, Jia Wang

October 2003 **Proceedings of the conference on Internet measurement conference**

Full text available:  pdf(236.92 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Per-flow traffic measurement is critical for usage accounting, traffic engineering, and anomaly detection. Previous methodologies are either based on random sampling (e.g., Cisco's NetFlow), which is inaccurate, or only account for the "elephants". Our paper introduces a novel technique for measuring per-flow traffic approximately, for all flows regardless of their sizes, at very high-speed (say, OC192+). The core of this technique is a novel data structure called Space Code Bloom Filter (SCBF). ...

Keywords: bloom filter, data structures, network measurement, statistical inference, traffic analysis

Results 1 - 20 of 139

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2003 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)